SEQUENTIAL DETECTION FOR A CLASS OF INFORMATION TRANSMISSION SYSTEMS

A Thesis Submitted
In Partial Fulfilment of the Requirements
For the Degree of

MASTER OF TECHNOLOGY

By
P. VIJAY KUMAR

ELECTRICAL ENGINEERING PROGRAM
INDIAN INSTITUTE OF TECHNOLOGY, KANPUR

AUGUST, 1979

CERTIFICATE

Certified that this work on 'SEQUENTIAL DETECTION FOR A CLASS OF INFORMATION TRANSMISSION SYSTEMS' by Shri P. Vijay Kumar has been carried out under my supervision and that it has not been submitted elsewhere for a degree.

PRKRao

Dr. P.R.K. Rao
Professor
Department of Electrical Engineering
Indian Institute of Technology
Kanpur

Kanpur
July 1979.

POST GRADUATE OFFICE
This thesis has been approved for the award of the Degree of Master of Technology (M.Tech.) in accordance with the regulations of the Indian Institute of Technology Kanpur
7.8.79 J

EE-1979-M-KUM-SEQ

CENTRAL LIBRARY
Acc. No. A 59510
14 SEP 1979

ACKNOWLEDGEMENTS

Firstly, I would like to thank Dr. P.R.K. Rao for the consistent confidence and guidance imparted throughout the duration of this research effort.

Thanks are also due to Dr. V. Sinha, Dr. P.K. Chatterjee and Dr. R. Subramaniam who alongwith Dr. P.R.K. Rao taught me the material forming the background of this thesis.

Finally, I would like to thank Shri S.S. Pethkar for an excellent and patient job of typing.

P. V. Kumar.

(P. VIJAY KUMAR)

ABSTRACT

In this thesis a sequential detection scheme is proposed for a class of information transmission systems (ITS) in which information regarding a particular message symbol is spread over several received symbols intervals. The channel for these ITSs is assumed to introduce additive white Gaussian noise (AWGN). Communication systems employing convolutional encoding or having inter symbol interference as well as feedback communication systems are cited as examples of such ITSs. Where a high error performance is desired fixed decoding delay detectors are commonly employed.

An interesting alternative considered here, is one in which a variable decoding delay is involved since only as many future received symbols (together with all the past received symbols) are examined as are necessary to make a decision regarding a message symbol with a certain level of confidence. A sequential detection test (SDT) is employed in this detector in order to determine as to when a satisfactory decision regarding a particular message symbol can be made.

Two sequential detectors employing in their SDTs a probability ratio and a maximum likelihood criterion respectively are considered in this report. The detector

employing a probability ratio for its SDT (the SDT is then referred to as the sequential probability ratio test SPRT) is shown to have an error performance comparable to that of the optimum fixed decoding delay receiver and hence a comparision between the two detectors is made revealing the sequential detector to enjoy a reduced computational complexity at the cost of a loss in decoding speed. However, both the detectors are in need of exponentiators for their implementation and hence an alternative sequential detector namely, the sequential detector employing a ML criterion is considered. This detector is shown to have an error performance comparable to that of the Viterbi detector and does away with the large survivor storage requirement of the Viterbi detector though it also suffers from a lower decoding speed. An extension of the generating function technique introduced by Viterbi is used to obtain bounds on the performance of this detector and these bounds are then evaluated for the specific case of communication systems employing convolutional coding. Simulation results are also presented, to determine the tightness of these bounds as well as to examine other important features of the sequential detector such as the applicability of decision feedback as well as the buffer requirement acruing from a variable decoding delay. Finally, a few conclusions as to when sequential detection can be applied are drawn on the basis of the above resul

CONTENTS

CONTENTS (contd.)

CHAPTER I

INTRODUCTION

1.1 The Topic of the Thesis and Related Work

This thesis is concerned with an application of the principle of sequential detection to the problems of reception arising in a certain class of digital information transmission systems.

In the information transmission systems (ITS) with which we are concerned in this thesis the problem is one of detection of a sequence of message symbols in situations wherein the information regarding any particular message symbol is spread out over several received symbols. For such receivers, the optimum receiver is one which examines all the received symbols before coming to a decision regarding the received symbol [CH- 66]. Since however this would lead to a receiver structure difficult to implement in practice one is restricted to processing a fixed number of received symbols before making a decision regarding a particular message symbol. The optimum receiver under such a fixed decoding constraint has already been considered (see [AB - 70], [CH - 66]). In this report, we are concerned

with an interesting alternative approach in which the principle of sequential detection is employed and only as many received symbols are examined as are necessary to make a decision regarding the message symbol with a certain predetermined level of confidence.

Sequential detection was introduced to the field of statistical decision theory by A. Wald [WAL-48], [WAL-47] in 1947. Its adoption to the theory of signal detection in communication systems dates back to almost the same year and various papers exist proposing the use of sequential detection to such topics as the detection of a sine wave in the presence of Gaussian noise (see Middleton and Bussgang [MB - 55]). In the year 1966, sequential detection was suggested as a means of minimizing the average number of requests for retransmission over the noiseless feedback channel of a feedback communication system by Turin [TUR-66] . Though a large number of papers have been published since then in this area, the idea has not been adopted in practice [LUC- 73]. The first major practical application of sequential detection to signal detection was perhaps in the field of target detection in Radar (see Di Franco and Rubin [DR - 68]), where it was shown to result in quicker target detection.

A characteristic of the class of information systems being considered in this chapter is that information regarding any particular received symbol is spread over several received symbols. To the best of the author's knowledge, no detection scheme, sequential or otherwise has been proposed for this class of information transmission systems, though various detection schemes have been suggested for specific examples of such ITS's such as the feedback communication system mentioned above. The two most frequently encountered examples of the ITS mentioned above are communication systems employing convolutional codes and communications systems in which a time dispersive channel leads to the presence of intersymbol interference (i.s.i.). Various techniques have been presented for the detection of message symbolsin both communication systems and we shall presently examine in brief a few techniques which are in common usage. Though the nature of the problem of signal detection in the two systems is the same i.e. involving the detection of message symbols in the case where several received symbols contain information relating to a particular message symbol, the decoding schemes in the two disciplines flourished independently whilst many of the techniques proposed for either one of the two systems are equally applicable to the other.

In the following we have a brief look at the three techniques commonly employed for the decoding of convolutional codes. These are feedback decoding introduced by Massey [MAS-63], sequential decoding introduced by Wozencraft [WOZ-57] and Viterbi decoding introduced by Viterbi [VIT-67]. All three techniques can with minor modifications be adapted for the detection of signals in the presence of i.s.i. The first of these - feedback decoding, is a decoding technique in which a decision regarding the present bit being decoded is made using past symbol decision feedback and only such received symbols as correspond to output symbols which are linear combinations of the past and present symbols.

This technique has the advantage of being simpler to implement though it is prone to error propogation effects and has an overall error performance considerably poorer than that of either of the two other techniques named above.

Sequential decoding is a decoding technique making use of the expected behaviour of the log likelihood function (given the received symbols) of the true message sequence in order to detect the message sequence. Using only a limited number of received symbols and assuming past decisions to be correct, the decoder follows a path in the code

tree making tentative decisions. The log likelihood or some other equivalent functions of the estimated message sequence is computed and continuously monitored. When the decoder finds the log likelihood function behaviour to be anamalous, it goes back on its previous decisions and traces a fresh path in the code tree. This decoding method has the disadvantage of requiring a large storage requirement as compared to the Viterbi decoder for small values K of the constraint length of the convolutional code. The reverse is the case for large values of K and the sequential decoding technique becomes attractive for values of $K > 10$ or so (see [VIT-71]). The Fano algorithm [FAN -63] detector which uses a tilted distance function in place of the above log likelihood function is the most commonly employed sequential decoding algorithm today.

The Viterbi algorithm detector described in [VIT -67], [VIT-71] and [HJ-71] is a detector estimating the i^{th} block of k message bits (where k/n is the rate of the convolutional code) of the message sequence considered as the most likely sequence (given the 1st $(i + N_T)$ blocks of n received symbols each) to be the true block of message symbols. For the case when $N_T = \infty$, the algorithm would clearly result in maximum likelihood sequence detection. However, in practice, a value of N_T

oughly equal to four times the constraint length of the code is commonly employed which value has been found to yield decisions having a probability of error nearly equal to that of the maximum likelihood sequence detector. However, the algorithm suffers from the disadvantage of having a storage requirement and computational complexity which grow exponentially with increase in the constraint length of the code. Whereas the computational complexity of the decoder can be reduced by using a serial implementation (see for e.g. [HJ- 71], [VB - 75]), the same is not true of the storage requirement.

In like fashion, several detection schemes have been proposed for the signals in the presence of intersymbol interference. The receivers currently being used to combat intersymbol intereference for the case when the pulse response of the channel is known, can be divided into the classes of linear and nonlinear equalizers. The optimal linear equalizer having a finite number of tap gains was shown by Tufts [TUF - 63] to consist of a matched filter followed by a transversal filer. The same structure is obtained even for suboptimal linear equalizers optimum either in the minimum mean square error sense or else optimum in the minimum probability of error sense under the constraint of having zero i.s.i. at the sampling instants.

However, linear equalizers have been found to have a poor error performance when compared to the statistically optimum receiver and it is this which lead to the current growing interest in suboptimal non linear equalizers. Of these, the decision feedback equalizer, the Viterbi algorithm receiver and the optimum fixed decoding delay receiver have received a considerable amount of attention in recent years.

The optimum fixed decoding delay receiver was first proposed by Chang and Hancock [CH - 66] whose receiver was optimum in the sense that it minimized the probability of error in detecting a sequence. Bowen [BOW - 69] and Abend et al. [AB - 68] pointed out that minimizing the probability of error in sequence detection was not equivalent to minimizing the probability of bit error and both suggested modifications in the receiver structure employed by Chang and Hancock. A modified receiver structure, alongwith simulation results obtained for the case of a particular tel. channel were then presented by Abend and Fritchman [AB - 70] . However, the optimum fixed delay constraint receiver needed the use of exponentiators thus making implementation difficult. Pointing out this fact, Austin [AUS - 67] considered a suboptimal equalizer employing decision feedback which was optimal

amongst all non-linear equalizers having the same structure. The receiver structure considered by Austin consisted of a forward filter operating upon future symbols and a backward filter making use of past decisions to eliminate past symbol interference. The forward and backward filters are transversal filters and the tap co-efficients of the forward filter were obtained by optimizing for the minimum probability of error under the assumption that future message symbols had a Gaussian amplitude distribution. Monsen [MON-71] and Storey [STO-71] adopted the receiver structure employed by Austin and rederived (independently) the optimal tap gains using a minimum mean square error criterion. No analysis of the error performance of the decision feedback equalizer (DFE) is available to date.

However, simulation results obtained by Monsen [MON - 74] indicate for the case of dispersive channels having an impulse response confined to a few baud intervals, the performance of the DFE to be not far inferior to that of the Viterbi algorithm detector.

The Viterbi algorithm detector was presented by Viterbi [VIT - 67] for the decoding of convolutional codes in 1967. A few years later, the idea of using the Viterbi algorithm for the detection of signals in the presence of intersymbol interference was proposed independently

by Forney [FOR - 72] , Kobayashi [KOB -71], [KOB-72] and Omura [OMU - 71] . However, Goutmann [GOU - 72] and Klein and Wolf [KW -71] had conceived some what earlier the idea of treating the problem of intersymbol interference using techniques borrowed from coding theory. Whereas Goutmann considered the application of the FANO decoding algorithm for the detection of signals in the presence of i.s.i., Klein and Wolf suggested the use of i.s.i. for error correction.

1.2 Organization of the Thesis

The thesis is organized as follows. In Chapter II, the principle of sequential detection is explained and the question regarding the applicability of sequential detection to communication systems raised. In partial answer to this query, an information transmission system (ITS) model to which sequential detection could be applied is presented. The ITS model considers an additive white Gaussian noise (AWGN) channel, though such a restriction is not necessary for sequential detection to be applicable. However in the case of any other channel, the applicability of the sequential detection techniques presented in this report must be examined afresh for each particular channel. The proceedure followed by a sequential detector for such an ITS is then outlined and finally few characteristics common to every sequential detector are discussed.

In Chapter III, a sequential detector employing the sequential probability ratio test (SPRT) is considered. The sequential detector is seen to have a comparable error performance to that of the optimum fixed decoding delay receiver but requiring for its implementation, a far lesser number of computational elements. As against this, the sequential detector has a lower decoding speed. The chapter is concluded with a presentation of a conservative bound on the bit error probability of the sequential detector.

In Chapter IV, a sequential detector employing a ML criterion is considered. The sequential detector is seen to have a comparable error performance to that of the Viterbi detector though requiring for its implementation a far lesser storage requirement. As against this, the sequential detector has a lower decoding speed. An extension of the generating function technique introduced by Viterbi is then employed to derive upper bounds on the probability of bit error as well as the Average Sample Number (ASN) of this sequential detector. The buffer requirement of the detector is also discussed and the chapter finally concluded by the presentation of extensive results obtained through evaluation of the bounds mentioned above as well as through simulation of the detector on a digital computer.

In the last Chapter of the thesis, Chapter V, the conclusions to be arrived at given the material of Chapters II, III and IV as well as the possibility of future work in this area are considered.

CHAPTER II

SEQUENTIAL DETECTION AND ITS APPLICATION TO DIGITAL COMMUNICATION SYSTEMS

In this Chapter, the principle of sequential detection is introduced and it is shown how one may employ sequential detection to detect data in a certain class of information transmission systems. In Section 1, the topic of sequential detection as applied to general statistical decision theory is dealt with. In Section 2, the possibility of applying sequential detection to a digital communication system is then considered and several examples of such communication systems where one might use sequential detection are mentioned. An information transmission system model to which sequential detection can be applied and to which all the communication systems mentioned above confirm is presented in Section 3. The procedure adopted by a sequential detection in the case of such an information transmission system is outlined in Section 4 together with a discussion on a few characteristics common to every sequential detector.

2.1 Sequential Detection.

Consider a situation in which one is

able to make several observations on the outcome of an experiment having one of two possible results. Let the observations be unlimited and also noisy in the sense that the observations are not quite sufficient to tell us definitely as to which event has occurred, but can only inform us in a probabilistic sense. Then briefly, sequential detection (see Wald [WAL-47]) is a technique which makes use of only as many observations as are necessary to make a decision regarding the outcome of the experiment with a certain degree of confidence. Whether or not the desired level of confidence has been attained is determined by subjecting the observations to a test which we shall refer to as the sequential detection test (SDT) . To illustrate the above, we now consider the example of target detection in Radar.

The antenna sweeps the sky repeatedly and each sweep brings in a return from the target region. Before sequential detection was introduced, the method commonly employed was one in which a fixed number N_T of returns were examined. On the basis of these, the decision as to the presence or absence of a target was made. Such a detector is called a fixed sample size (FSS) detector.

In the case of sequential detection, the observation space is divided into three regions:-

A region R_0 in which the target is declared to be absent,

A region R_1 in which the target is declared to be present

and A region R_c in which the next return is awaited.

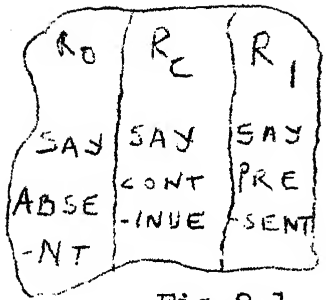


Fig.2.1 Observation Space

The receiver examines the first return and either makes or does not make a decision depending upon the region of the observation space in which the return happens to lie. If no decision is made, the second return is examined and so on.

It has been theoretically proved that the average number of returns from the target region examined by the sequential detector in such a case, is lesser for a given probability of target detection as compared to the fixed sample size (FSS) receiver mentioned earlier, thus resulting in quicker target detection. For a more detailed description of this topic the interested reader is referred to Difranco and Rubin [DR - 68] .

2.2 Applicability of Sequential Detection to Digital Communication Systems.

Now, in a digital communication system, one is in general, concerned with the detection of a message symbol on the basis of one or more than one received symbols. It is apparent that sequential detection can be applied only in the latter case. Several examples of such communication

systems come to mind such as the detection of signals in the presence of intersymbol interference, the decoding of convolutionally encoded data and the detection of signals in the presence of a feedback channel using the automatic repeat request (ARQ) scheme. Its application to the last mentioned of these was first suggested by Turin [TUR-66] In each of these cases, information regarding any specific message symbol is spread over a span of several received symbols.

Rather than discuss the application of sequential detection to each of the above systems in turn, we will consider instead an information transmission system model to which each of the above systems can be reduced. We will then apply ourselves to the far simpler task of describing the application of sequential detection to this model.

2.3 The Information Transmission System Model.

The information transmission system model which we consider in this Chapter consists of a finite state machine (FSM) encoding the output of a message source, a modulator mapping the output alphabet of the FSM into a set of real valued parameters suitable for transmission, a channel introducing additive white Gaussian noise and a receiver making an estimate of the message sequence based on

the channel output. (see Fig. 2.2). For the sake of clarity in description, we shall confine our attention here to the class of FSM's having the binary input alphabet [0, 1]. Such a machine may be represented by the 5-tuple $\langle I, Q, Z, \delta, w \rangle$ (see [BTH - 67]) where I and Z denote the input and output alphabet, Q the set of states and where δ and w denote the next state and output mapping functions respectively.

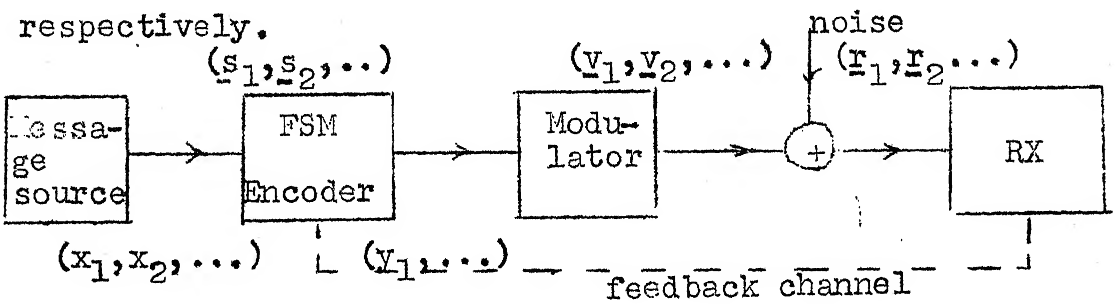


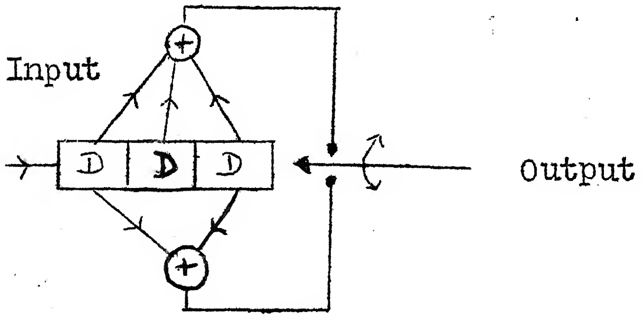
Fig. 2.2 Information Transmission System Model.

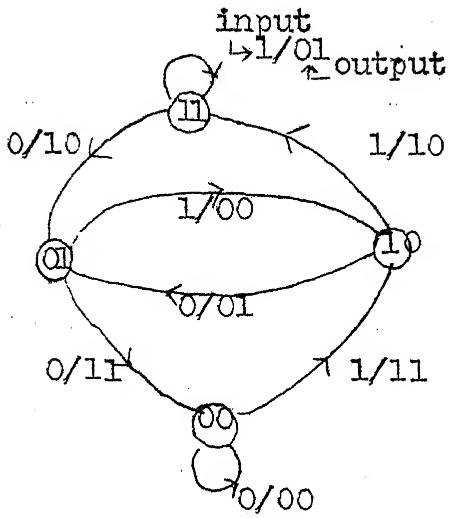
Figures 2.3 a, b and c show the state diagrams of the FSM which will be used to model the information transmission system (ITS) in the case of a convolutional encoder, a time dispersive channel and a transmitter employing ARQ respectively. The state diagram for the case of convolutional codes is self explanatory. In the case of ARQ, a slight modification must be made in the information transmission system and a feedback channel having the ability to control the state of the FSM introduced between the transmitter and the receiver as indicated by the dotted

lines in Fig. 2.2. In this scheme, the transmitter keeps retransmitting the latest symbol for as long as a request for repeat is maintained. In the absence of such a request of course, the current message symbol is passed on to the modulator. States B0 and B1 represent the machine in its retransmit mode whilst states A0 and A1 correspond to the normal mode of transmission. The input in this state diagram is a 2 component vector representing data received from both the message source and over the feedback channel.

Whilst past data are being retransmitted, the FSM keeps the input from the message source pending. This explains therefore the presence of only three symbols in the input alphabet of the FSM:- $\left[\binom{0}{0}, \binom{1}{0} \text{ and } \binom{\emptyset}{1} \right]$, where $\emptyset$ indicates an absence of message data during retransmission. Of course, the loss in time due to retransmission is made up in the interval between transmission bursts.

In the case of i. s. i, we use the FSM and modulator to reproduce the signal at the output of the whitened matched filter (see Forney [FOR-72]) at the receiver end. The signal consists of a sequence of discrete time samples in which each received sample is dependent upon the corresponding message symbol as well as those prior to it. In the model we have shown, each sample is a function of the present and immediate past inputs to the FSM and this





$$I = [0,1]$$

$$Q = [00,\ 01,\ 10,\ 11]$$

$$Z = [00,\ 01,\ 10,\ 11]$$

$$\underline{z}_k = \begin{bmatrix} z_{k1} \\ z_{k2} \end{bmatrix}$$

$$\underline{v}_k = \begin{bmatrix} v_{k1} \\ v_{k2} \end{bmatrix}$$

$$v_{ki} = h\,\alpha_{ki}$$

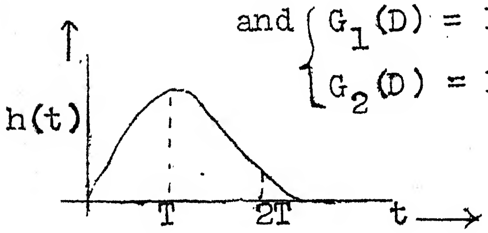
where :- h is a constant,

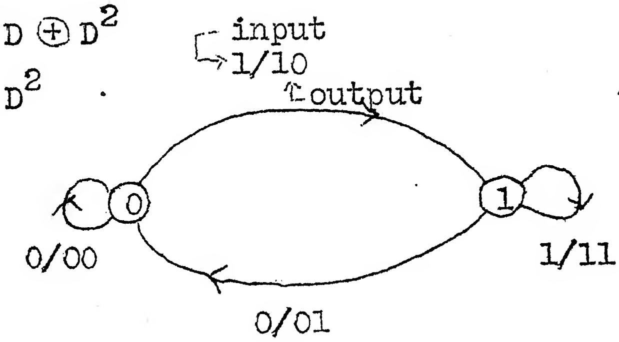
$$\alpha_{ki} = +1 \text{ if } z_{ki} = 1,$$

$$\alpha_{ki} = -1 \text{ if } z_{ki} = 0$$

$$i = 1, 2,$$

Fig. 2.3a Rate 1/2 Convolutional Encoder with K= 3 and $\begin{cases} G_1(D) = 1 \oplus D \oplus D^2 \\ G_2(D) = 1 \oplus D^2 \end{cases}$.





T = baud interval

$$q_k = i_{k-1}$$

$$\underline{z}_k = \begin{bmatrix} z_{k1} \\ z_{k2} \end{bmatrix}$$

$$v_k = h_0\,\alpha_{k1} + h_1\,\alpha_{k2}$$

where h_0, h_1 are constants

$$\alpha_{ki} = +1, \text{ if } z_{ki} = 1$$

$$\alpha_{ki} = -1, \text{ if } z_{ki} = 0 \qquad i=1,2.$$

Fig. 2.3b. ISI model for the case when impulse response spreads over 2 baud intervals.

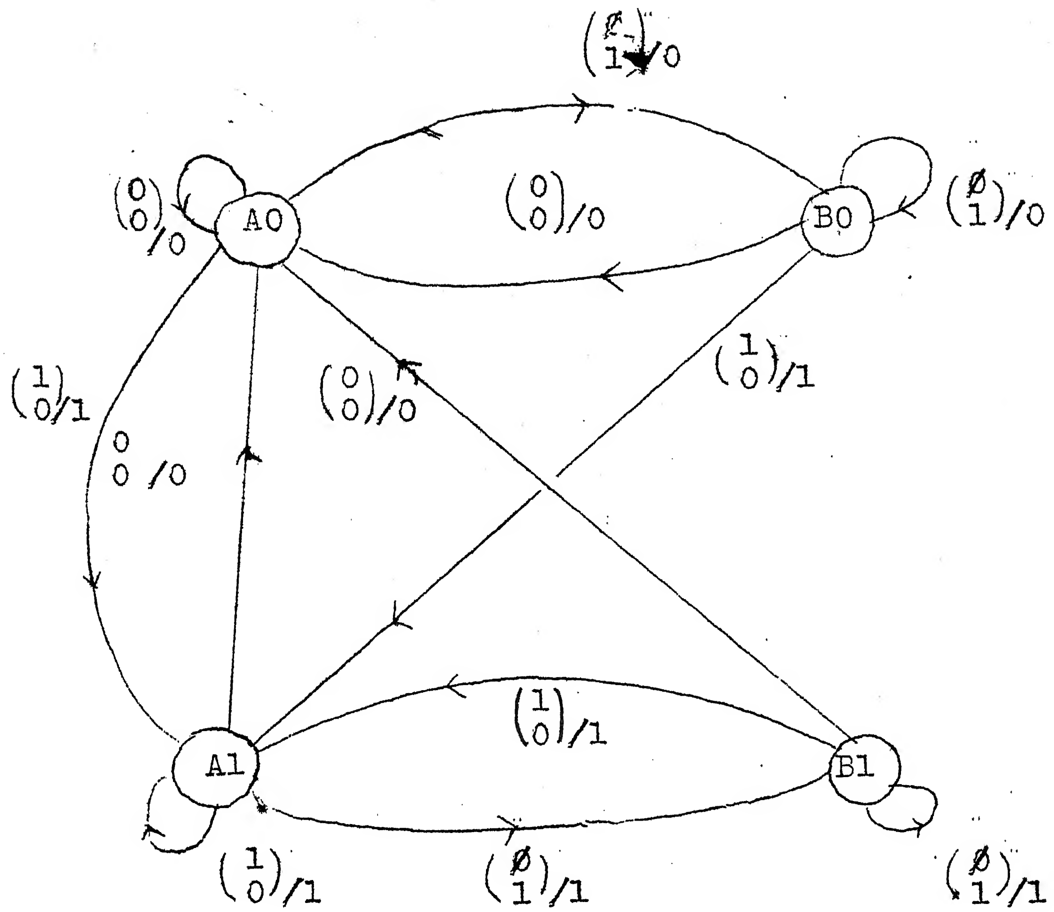


Fig. 2.3c FSM model for transmitter employing ARQ.

where ; :

$$I = \left[\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} \emptyset \\ 1 \end{pmatrix} \right]$$

$$Z = [0,1]$$

$v_k = h\,\alpha_k$ where $\alpha_k = +1$ if $z_k = +1$

h is a constant. $= -1$ if $z_k = 0$.

corresponds to the case of a channel having a pulse response extending over two baud intervals. The relative values of h_0 and h_1 can be shown to be indicative of the relative signal energy in the first and second baud intervals.

1.4 Sequential Detection as applied to the ITS Model.

We will in this Section describe the sequential algorithm as applied to the ITS model just discussed and then go on to consider a few features of sequential detection. The algorithm proceeds in such a way that prior to the detection of the message symbol x_i, the symbols $\underline{r}_1$, $\underline{r}_2$, ... , $\underline{r}_{i-1}$ have already been processed and the result stored. The first step is therefore processing the symbol $\underline{r}_i$, after which the sequential detection test (SDT) is applied. As in the case of target detection in Radar, at any stage in the detection process, the observation region is divided into three regions:-

A region R_o in which x_i is declared to be 0

A region R_1 in which x_i is declared to be 1

A region R_c in which no decision is reached and the next received symbol is examined.

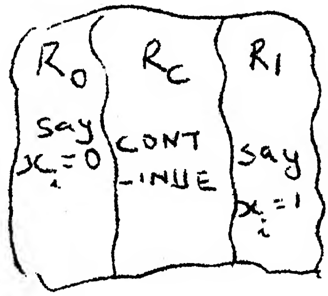


Fig. 3.4

Observation Space

Hence, depending upon the region of the observation space in which the received symbols happen to lie, either a

decision regarding the symbol x_i is made or else, the next received symbol $\underline{r}_{i+1}$ is examined. This process is then repeated with the symbols $\underline{r}_{i+1}$, $\underline{r}_{i+2}$...in succession. (We note here that the processed result after the symbol $\underline{r}_{i+1}$ has been examined is stored to initiate the detection process for the next bit x_{i+1}.) An upper limit of $(N_T + 1)$ future symbols are examined beyond which the test is forcibly terminated since it is possible that the sequential detector may in some cases, observe a very large number of future received symbols without being able to satisfy the sequential detection test.

If we denote by p(j) the probability that the test will terminate after the observation of the j^{th} received symbol, then the average number of future symbols observed (considering $\underline{r}_i$ to be the first future symbol) is given by $\sum_{j=i}^{i+N_T} (j - i+1) \cdot p(j)$. We will refer to this quantity as the average sample number (ASN). Hence we have

$$ASN = \sum_{j=i}^{i+N_T} (j - i+1) \cdot p(j) \qquad \ldots (2.1)$$

Since there is a variable decoding delay in the sequential detection process it follows that a buffer is required both at the input and the output of the receiver. Whereas, the input buffer requirement arises from the fact that the incoming data arrive at constant intervals

of time, but are processed with a varying delay, the output buffer is needed since decisions are made irregularly but need to be put out at constant intervals of time.

The importance of the ASN may readily be appreciated from the following :

Let n_R be the maximum number of received symbols that can be processed by the receiver in 1 sec. Then we see that the maximum data rate n_D at which we can operate is given by

$$n_D \leq (n_R / \text{ASN}) \qquad \ldots (2.2)$$

Hence we see that the ASN determines the maximum bit rate capability of the sequential detector.

With this, we conclude our description of the sequential detection algorithm. We have deliberately not specified the nature of the SDT to be employed since many different tests could be used with varying consequences and indeed in Chapters III and IV we deal with sequential detection using two different sequential detection tests.

CHAPTER III

SEQUENTIAL DETECTION USING THE SEQUENTIAL PROBABILITY RATIO TEST

In this Chapter we consider for the information transmission system model outlined in Chapter 2, a sequential detector using the sequential probability ratio test (SPRT). To begin with, we examine in Section 1 the optimal fixed delay constraint receiver for the ITS model and show how it may be implemented in two ways one of which is recursive and the other semi-recursive. It turns out that of the two procedures, the one which is semirecursive in nature is the one that is simpler to implement though having a far lower decoding speed. This receiver which we consider in Section 2, is analogous to the fixed sample size detector mentioned earlier in the example of target detection in Radar. We modify the structure of this receiver using sequential detection instead of FSS detection and obtain a receiver structure which whilst sharing the reduced complexity of the FSS detector has a considerably greater maximum bit rate capability. In Section 3, we show how such a detector may be implemented in practice and then go on to present in Section 4 an upper bound on the bit error performance of this detector. Finally, we conclude in Section 5 with a discussion

on the relative merits and demerits of such a sequential detector.

3.1. The Optimum Fixed Decoding Delay Receiver.

The optimum receiver (in the minimum probability of error sense) for the ITS considered in this report is one which evaluates the ratio

$$\frac{\Pr\left(H_i^0 / \underline{r}_1, \underline{r}_2, \ldots, \underline{r}_\infty \right)}{\Pr\left(H_i^1 / \underline{r}_1, \underline{r}_2, \ldots, \underline{r}_\infty \right)}$$

in order to detect the message symbol x_i and declares x_i to be a 0 or a 1 depending upon whether the ratio is greater or lesser than unity. Here H_i^0 and H_i^1 stand for the hypothesis that the i^{th} message symbol is a 0 and a 1 respectively. Such a receiver is clearly not possible to implement in practice and hence we are led to consider the optimum receiver operating under a fixed finite decoding delay constraint D. By a fixed decoding delay D, we mean that an interval of D baud intervals elapses between the arrival of the symbol $\underline{r}_i$ and the detection of the symbol x_i. Such a receiver would clearly evaluate the ratio:-

$$\frac{\Pr\left(H_i^0 / \underline{r}_1, \underline{r}_2, \ldots, \underline{r}_{i+D} \right)}{\Pr\left(H_i^1 / \underline{r}_1, \underline{r}_2, \ldots, \underline{r}_{i+D} \right)}$$

before making a decision on the symbol x_i as above.

Alternatively, since

$$p(\underline{r}_1, \underline{r}_2, \ldots\ldots \underline{r}_{i+D} / H_i^o) = \frac{Pr(H_i^o / \underline{r}_1, \underline{r}_2, \ldots, \underline{r}_{i+D})}{Pr(H_i^o)} \times$$

$$\times\; p(\underline{r}_1, \underline{r}_2, \ldots, \underline{r}_{i+D})$$

and we assume that the a priori probabilities of the symbols 0 and 1 are equal , the receiver could equivalently evaluate the ratio

$$\frac{p(\underline{r}_1, \underline{r}_2, \ldots\ldots, \underline{r}_{i+D} / H_i^o)}{p(\underline{r}_1, \underline{r}_2, \ldots\ldots, \underline{r}_{i+D} / H_i^1)} .$$

Such a receiver has been considered by Abend and Fritchman [AB - 70] for the particular case of a channel having intersymbol interference. The following material on this optimum receiver is an adaption of the recursive structure employed by them to the more general ITS considered in this report. The structure is valid only for ITS's in which the FSM is such that its output is a function only of a finite number 'L' of the immediate past symbols fed to it by the message source. We will therefore confine ourselves to such ITS's in this Chapter. Also as one would logically expect, Abend and Fritchman obtained simulation results indicating that a choice of D < L resulted in a severe loss in performance

as compared to choices of $D \geqslant L$ and we shall therefore consider the latter only in this report.

The recursive structure considered by Abend and Fritchman for such a receiver is described below. It is assumed throughout that the message symbols are independent of each other and that their a priori probabilities are equal. We have,

$$p(\underline{r}_1, \underline{r}_2, \ldots \underline{r}_{i+D} / H_i) =$$

$$\sum_{\text{all } H_{i+1}, H_{i+2}, \ldots, H_{i+D}} p(\underline{r}_1, \underline{r}_2, \ldots, \underline{r}_{i+D} / H_i, \ldots, H_{i+D}) \times \Pr(H_{i+1}, H_{i+2}, \ldots, H_{i+D}) \qquad \ldots(3.1)$$

Also,

$$p(\underline{r}_1, \ldots, \underline{r}_{i+D}/H_i, \ldots, H_{i+D}) = \sum_{\text{all } H_{i-1}} p(\underline{r}_1, \ldots, \underline{r}_{i+D}/H_{i-1}, \ldots, H_{i+D}) \times \Pr(H_{i-1}).$$

$$= p(\underline{r}_{i+D} / H_{i+D-L}, \ldots, H_{i+D}) \cdot \frac{1}{2} \cdot \sum_{\text{all } H_{i-1}} p(\underline{r}_1, \ldots, \underline{r}_{i+D-1} / H_{i-1}, \ldots, H_{i+D-1}) \qquad \ldots(3.2)$$

where $H_i, \ldots, H_{i+D}$ stands for a hypothesis regarding the sequence $x_i, x_{i+1}, \ldots, x_{i+D}$. Hence, we have obtained a means of recursively evaluating $p(\underline{r}_1, \ldots, \underline{r}_{i+D}/H_i, \ldots, H_{i+D})$ for $i = 1, 2, \ldots$ for all possible hypotheses $H_{i+1}, \ldots, H_{i+D}$ and

these can be used in equation (3.1) to obtain $p(\underline{r}_1, .., \underline{r}_{i+D}/H_i^0)$ and $p(\underline{r}_1, ..., \underline{r}_{i+D}/H_i^1)$ and hence the probability ratio needed for the decision to be made.

To illustrate this method of computation, we will rewrite equations (3.1) and (3.2) in vector form for the particular case L = 1 , D = 3. (see equations (3.3) to (3.5)).

We denote :

by $\underline{QA}(i)$, the conditional density function vector on the LHS of eqn. (3.3a)

by $\pi A(i)$, the transition matrix in eqn. (3.3a)

by $\pi A_m(i)$ and πA_a, the two matrices into which $\pi A(i)$ is split,

i.e. $$\pi A(i) = \pi A_m(i) \cdot \pi A_a \qquad ...(3.6)$$

by $\underline{PA}(i)$, the conditional density function vector on the LHS of eqn. (3.5a)

by ∂A , the transition matrix in eqn. (3.5b)

and by $\underline{SA}(i)$, the vector $= \frac{1}{2} \cdot \pi A_a \cdot \underline{QA}(i-1)$

i.e. $$\underline{SA}(i) = \frac{1}{2} \cdot \pi A_a \cdot \underline{QA}(i-1) \qquad ...(3.7)$$

To detect the i^{th} message symbol, the receiver proceeds as follows:-

$$
\begin{bmatrix}
(\underline{r}_1,\ldots,\underline{r}_{i+3}/H_i^0\ H_{i+1}^0\ H_{i+2}^0\ H_{i+3}^0) \\
(\underline{r}_1,\ldots,\underline{r}_{i+3}/H_i^0\ H_{i+1}^0\ H_{i+2}^0\ H_{i+3}^1) \\
(\underline{r}_1,\ldots,\underline{r}_{i+3}/H_i^0\ H_{i+1}^0\ H_{i+2}^1\ H_{i+3}^0) \\
\vdots \\
p(\underline{r}_1,\ldots,\underline{r}_{i+3}/H_i^1\ H_{i+1}^1\ H_{i+2}^1\ H_{i+3}^1)
\end{bmatrix}
= \frac{1}{2}
\begin{bmatrix}
f_o & & & & & & & & f_o & & & & & & & \\
f_1 & & & & & & & & f_1 & & & & & & & \\
& f_2 & & & & & & & & f_2 & & & & & & \\
& f_3 & & & & & & & & f_3 & & & & & & \\
& & f_o & & & & & & & & f_o & & & & & \\
& & f_1 & & & & & & & & f_1 & & & & & \\
& & & f_2 & & & & & & & & f_2 & & & & \\
& & & f_3 & & & & & & & & f_3 & & & & \\
& & & & f_o & & & & & & & & f_o & & & \\
& & & & f_1 & & & & & & & & f_1 & & & \\
& & & & & f_2 & & & & & & & & f_2 & & \\
& & & & & f_3 & & & & & & & & f_3 & & \\
& & & & & & f_o & & & & & & & & f_o & \\
& & & & & & f_1 & & & & & & & & f_1 & \\
& & & & & & & f_2 & & & & & & & & f_2 \\
& & & & & & & f_3 & & & & & & & & f_3
\end{bmatrix}
\begin{bmatrix}
p(\underline{r}_1,\ldots,\underline{r}_{i+2}/H_{i-1}\ H_i\ H_{i+1}\ H_{i+2}) \\
p(\underline{r}_1,\ldots,\underline{r}_{i+2}/H_{i-1}^0\ H_i^0\ H_{i+1}^0\ H_{i+2}^1) \\
p(\underline{r}_1,\ldots,\underline{r}_{i+2}/H_{i-1}^0\ H_i^0\ H_{i+1}^1\ H_{i+2}^0) \\
\vdots \\
p(\underline{r}_1,\ldots,\underline{r}_{i+2}/H_{i-1}^1\ H_i^1\ H_{i+1}^1\ H_{i+2}^1)
\end{bmatrix}
\qquad \ldots\ (3.3a)
$$

$$
\text{i.e.} \quad \underline{QA}\,(i) = \pi A\,(i)\,.\,\underline{QA}\,(i-1) \qquad \ldots\ (3.3b)
$$

$$
f_o = p(\underline{r}_{i+3}/H_{i+2}^0\ H_{i+3}^0),\ f_1 = p(\underline{r}_{i+3}/\ H_{i+2}^0\ H_{i+3}^1)\ ,\ f_2 = p(\underline{r}_{i+3}/H_{i+2}^1\ H_{i+3}^0),\ f_3 = p\,(\underline{r}_{i+3}/H_{i+2}^1\ H_{i+3}^1)
$$

(3.3c)

$$\pi A(i)=\begin{bmatrix}
f_0&&&&&&&&&&&&&&&\\
&f_1&&&&&&&&&&&&&&\\
&&f_2&&&&&&&&&&&&&\\
&&&f_3&&&&&&&&&&&&\\
&&&&f_0&&&&&&&&&&&\\
&&&&&f_1&&&&&&&&&&\\
&&&&&&f_2&&&&&&&&&\\
&&&&&&&f_3&&&&&&&&\\
&&&&&&&&f_0&&&&&&&\\
&&&&&&&&&f_1&&&&&&\\
&&&&&&&&&&f_2&&&&&\\
&&&&&&&&&&&f_3&&&&\\
&&&&&&&&&&&&f_0&&&\\
&&&&&&&&&&&&&f_1&&\\
&&&&&&&&&&&&&&f_2&\\
&&&&&&&&&&&&&&&f_3
\end{bmatrix}
\cdot
\begin{array}{c}
\begin{array}{cccccccccccccccc}1&&&&&&&&9&&&&&&&\end{array}\\
\begin{bmatrix}
1&&&&&&&&1&&&&&&&\\
1&&&&&&&&1&&&&&&&\\
&1&&&&&&&&1&&&&&&\\
&1&&&&&&&&1&&&&&&\\
&&1&&&&&&&&1&&&&&\\
&&1&&&&&&&&1&&&&&\\
&&&1&&&&&&&&1&&&&\\
&&&1&&&&&&&&1&&&&\\
&&&&1&&&&&&&&1&&&\\
&&&&1&&&&&&&&1&&&\\
&&&&&1&&&&&&&&1&&\\
&&&&&1&&&&&&&&1&&\\
&&&&&&1&&&&&&&&1&\\
&&&&&&1&&&&&&&&1&\\
&&&&&&&1&&&&&&&&1\\
&&&&&&&1&&&&&&&&1
\end{bmatrix}
\end{array}$$

... (3.4a)

i.e. $\pi A\ (i) = \pi A_m(i)\ .\ \pi A_a$ ... (3.4b)

$$\begin{bmatrix} p(\underline{r}_1,\ldots,\underline{r}_{i+3}/H_i^0) \\ p(\underline{r}_1,\ldots,\underline{r}_{i+3}/H_i^1) \end{bmatrix} = \begin{array}{c}
\begin{array}{cccccccccccccccc}1&&&&&&&&9&&&&&&&\end{array}\\
\begin{bmatrix} 1&1&1&1&1&1&1&1&&&&&&&&\\ &&&&&&&&1&1&1&1&1&1&1&1 \end{bmatrix}\end{array} \cdot \underline{QA}\ (i)$$

... (3.5a)

i.e. $\underline{PA}(i) = \partial A\ .\ \underline{QA}\ (i)$ ... (3.5b)

Step 1

The conditional density function vector $\underline{SA}(i)$ giving the conditional density functions $p(r_1, r_2, \ldots, r_{i+D-1}/H_i\ H_{i+1} \ldots H_{i+D-1})$ for all possible combinations of $H_i\ H_{i+1} \ldots H_{i+D-1}$ is available from the previous calculation . The first step is then the evaluation of the elements $f_o\ f_1\ f_2\ f_3$ which for the additive Gaussian noise channel we are considering will involve the use of exponentiators.

Step 2

Evaluation of the conditional density functions comprising the vector $\underline{QA}(i)$ as follows :-

$$\underline{QA}(i) = \pi A_m(i) \cdot \left[\frac{1}{2} \cdot \pi A_a(i) \cdot \underline{QA}(i-1)\right] = \pi A_m(i) \cdot \underline{SA}\ (i)$$

This matrix multiplication is a process in which one clearly requires the use of 16 multipliers if parallel implementation is desired. In general one would need 2^{D+1} multipliers.

Step 3

Computation of the conditional density functions comprising the vector $\underline{PA}\ (i)$ according to the equation :-

$$\underline{PA}(i) = \partial A(i) \cdot \underline{QA}(i)$$

This clearly involves two additions to be performed and the number of additions is invariant with respect to L and D.

Step 4

Using the elements of $\underline{PA}(i)$, one computes the probability ratio :-

$$\frac{p(\underline{r}_1, \underline{r}_2, \ldots, \underline{r}_{i+3}/H_i^0)}{p(\underline{r}_1, \underline{r}_2, \ldots, \underline{r}_{i+3}/H_i^1)}$$

If the ratio is greater than 1, the symbol x_i is declared to be a '0' otherwise a '1'.

Step 5

To obtain and store the conditional density functions needed for the detection of the symbol x_{i+1}. To do this we evaluate:-

$$\underline{SA}\ (i+1) = \frac{1}{2}\ \pi A_a \cdot \underline{QA}(i)$$

which is a process involving 2^{D+1} additions. In general, the vector $\underline{SA}$ (i+1) will comprise of 2^{D+1} terms (16 in this case) of which only 2^D are distinct as is obvious from the nature of πA_a .

This proceedure is then repeated with succeeding symbols. The reason for splitting up the matrix $\pi A(i)$ is to bring out clearly the number of multipliers , adders as well as the amount of storage needed. In general, one would need 2^{D+1} multipliers , $(2^{D+1} + 2)$ adders, 2^{L+1} exponentiators

and storage place for 2^D analog samples in order to implement this receiver.

3.2. An Alternative Implementation of the Optimum Fixed Delay Receiver.

We will now consider an alternative proceedure for evaluating the density function ratio

$$\frac{p(\underline{r}_1, \underline{r}_2, \ldots, \underline{r}_{i+D}/H_i^0)}{p(\underline{r}_1, \underline{r}_2, \ldots, \underline{r}_{i+D}/H_i^1)} \qquad \text{for } i = 1,2,\ldots .$$

The proceedure is semi recursive in the sense that whilst the processed result of past received symbols $(\ldots, \underline{r}_{i-2}, \underline{r}_{i-1})$ is used directly, the future symbols $(\underline{r}_i, \underline{r}_{i+1}, \ldots, \underline{r}_{i+D})$ are examined afresh before the density function ratio indicated above is evaluated.

We begin by assuming that the conditional density functions $p(\underline{r}_1, \underline{r}_2, \ldots, \underline{r}_i/H_{i-L}, H_{i+1-L}, \ldots, H_i)$ for all combinations of hypotheses H_{i-1} $H_{i+1-L} \ldots H_i$ are available to us from a previous computation. Then, we proceed to evaluate recursively $p(\underline{r}_1, \underline{r}_2, \ldots, \underline{r}_{i+j}/H_i H_{i+j-L} \ldots H_{i+j})$ for $j = 1, 2, \ldots, D$ as follows :

$$p(\underline{r}_1, \underline{r}_2, \ldots, \underline{r}_{i+j}/H_i, H_{i+j-L} \ldots H_{i+j})$$

$$= \sum_{\text{all } H_{i+j-L-1}} p(\underline{r}_1, \ldots, \underline{r}_{i+j-1}/H_i, H_{i+j-L-1} \ldots H_{i+j-1}) \cdot$$

$$\times \quad p(\underline{r}_{i+j}/H_i, H_{i+j-L} \ldots H_{i+j})$$

$$\times \quad \Pr(H_{i+j-L-1}/H_i, H_{i+j-L} \ldots, H_{i+j}) \qquad \ldots (3.7)$$

Clearly, $\Pr(H_{i+j-L-1} / H_i, H_{i+j-L} \ldots H_{i+j})$

$= \Pr(H_{i+j-L-1}/H_i, H_{i+j-L} \ldots H_{i+j-1})$ since $j \geqslant 1$, which fact we shall use in the example to follow. This factor $= \frac{1}{2}$ or 0 depending upon the compatibility of the set of hypotheses $(H_i, H_{i+j-L-1} \ldots H_{i+j-1})$.

The conditional density function $p(\underline{r}_1, \ldots, \underline{r}_{i+j}/H_i)$ can be evaluated as :-

$$p(\underline{r}_1, \ldots, \underline{r}_{i+j}/H_i) = \sum_{\substack{\text{all} \\ H_{i+j-L} \ldots H_{i+j}}} p(\underline{r}_1, \ldots, \underline{r}_{i+j}/H_i, H_{i+j-L} \ldots H_{i+j})$$

$$\times \quad \Pr(H_{i+j-L} \ldots H_{i+j}/H_i) \qquad \ldots (3.8)$$

As above, the value of $\Pr(H_{i+j-L} \ldots H_{i+j}/H_i)$ will be either $\frac{1}{2^{L+1}}$, $\frac{1}{2^L}$ or 0 depending upon whether $j \geqslant L$ and also upon the compatibility of the set of hypotheses

$(H_{i+j-L} \ldots H_{i+j}, H_i)$. However, whenever this factor is a 0, the conditional density function multiplying it in equation (3.8) will also be $= 0$ and hence we can without loss in accuracy, put this factor $= \frac{1}{2^{S+1}}$ always where

$$
\begin{aligned}
S &= L-1 \qquad && j \leq L \\
&= L && \text{otherwise.}
\end{aligned}
$$

The conditional density functions needed to begin the detection process for the next bit $\bar{x}_{i+1}$, can be evaluated as follows :-

$$
\begin{aligned}
&p(\underline{r}_1, \ldots, \underline{r}_{i+1} / H_{i+1-L} \ldots H_{i+1}) \\
&= \sum_{\text{all } H_i} p(\underline{r}_1, \ldots, \underline{r}_{i+1} / H_{i+1-L} \ldots H_{i+1}, H_i) \times \Pr(H_i / H_{i+1-L} \ldots H_{i+1})
\end{aligned}
\qquad \ldots (3.9)
$$

Here again, we note that the factor $\Pr(H_i / H_{i+1-L} \ldots H_{i+1})$ will be $= 1$ or 0 depending upon the compatibility of the set of hypotheses $(H_i, H_{i+1-L} \ldots H_{i+1})$. For the identical reason put forward above, we can set this quantity also $= 1$ without loss in accuracy.

The equations (3.10) to (3.15) shown below illustrate the computations indicated by equations (3.7) to (3.9) for the particular case of $D = 3$, $L = 1$ also considered in the previous section.

$$
\begin{bmatrix}
p(\underline{r}_1,\ldots,\underline{r}_{i+j}/H_i^0 H_{i+j-1}^0 H_{i+j}^0) \\
p(\underline{r}_1,\ldots,\underline{r}_{i+j}/H_i^0 H_{i+j-1}^0 H_{i+j}^1) \\
p(\underline{r}_1,\ldots,\underline{r}_{i+j}/H_i^0 H_{i+j-1}^1 H_{i+j}^0) \\
\cdot \\ \cdot \\ \cdot \\ \cdot \\
p(\underline{r}_1,\ldots,\underline{r}_{i+j}/H_i^1 H_{i+j-1}^1 H_{i+j}^1)
\end{bmatrix}
=
\begin{bmatrix}
q_o f_o & & q_2 f_o & & & & & \\
q_o f_1 & & q_2 f_1 & & & & & \\
& q_1 f_2 & & q_3 f_2 & & & & \\
& q_1 f_3 & & q_3 f_3 & & & & \\
& & & & q_4 f_o & & q_6 f_o & \\
& & & & q_4 f_1 & & q_6 f_1 & \\
& & & & & q_5 f_2 & & q_7 f_2 \\
& & & & & q_5 f_3 & & q_7 f_3
\end{bmatrix}
\cdot
\begin{bmatrix}
p(\underline{r}_1,\ldots,\underline{r}_{i+j-1}/H_i^0 H_{i+j-2}^0 H_{i+j-1}^0) \\
p(\underline{r}_1,\ldots,\underline{r}_{i+j-1}/H_i^0 H_{i+j-2}^0 H_{i+j-1}^1) \\
p(\underline{r}_1,\ldots,\underline{r}_{i+j-1}/H_i^0 H_{i+j-2}^1 H_{i+j-1}^0) \\
\cdot \\ \cdot \\ \cdot \\ \cdot \\
p(\underline{r}_1,\ldots,\underline{r}_{i+j-1}/H_i^1 H_{i+j-2}^1 H_{i+j-1}^1)
\end{bmatrix}
\qquad \ldots (3.10a)
$$

$$
\text{i.e.} \quad \underline{QB_i}(j) = \pi B_i(j) \cdot \underline{QB_i}(j-1) \qquad \ldots (3.10b)
$$

$$
\pi B_i(j) =
\begin{bmatrix}
f_o & & & & & & & \\
& f_1 & & & & & & \\
& & f_2 & & & & & \\
& & & f_3 & & & & \\
& & & & f_o & & & \\
& & & & & f_1 & & \\
& & & & & & f_2 & \\
& & & & & & & f_3
\end{bmatrix}
\begin{array}{c}
\begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array} \\
\begin{bmatrix}
q_o & & q_2 & & & & & \\
q_o & & q_2 & & & & & \\
& q_1 & & q_3 & & & & \\
& q_1 & & q_3 & & & & \\
& & & & q_4 & & q_6 & \\
& & & & q_4 & & q_6 & \\
& & & & & q_5 & & q_7 \\
& & & & & q_5 & & q_7
\end{bmatrix}
\end{array}
\qquad \ldots (3.11a
$$

$$\text{i.e.} \quad \pi B_i(j) = \pi B_{i,m}(j) \cdot \pi B_{i,a}(j) \qquad \ldots(3.11b)$$

$$\begin{bmatrix} p(\underline{r}_1,\ldots,\underline{r}_{i+j}/H_i^0) \\ p(\underline{r}_1,\ldots,\underline{r}_{i+j}/H_i^1) \end{bmatrix} = \frac{1}{2^{S+1}} \cdot \begin{array}{c} \underline{1} \quad 2 \quad \underline{3} \\ \begin{bmatrix} 1111 & \\ & 1111 \end{bmatrix} \end{array} \cdot \underline{QB}_i(j) \qquad \ldots(3.12a)$$

$$\text{i.e.} \quad \underline{PB}_i(j) = \partial B \cdot \underline{QB}_i(j) \qquad \ldots(3.12b)$$

$$f_0 = p(\underline{r}_{i+j}/H^0_{i+j-1}H^0_{i+j}), \quad f_1 = p(\underline{r}_{i+j}/H^0_{i+j-1}H^1_{i+j}),$$

$$f_2 = p(\underline{r}_{i+j}/H^1_{i+j-1}H^0_{i+j}), \quad f_3 = p(\underline{r}_{i+j}/H^1_{i+j-1}H^1_{i+j}) \qquad \ldots(3.13)$$

$$q_0 = Pr(H^0_{i+j-2}/H^0_iH^0_{i+j-1}) \qquad q_4 = Pr(H^0_{i+j-2}/H^1_iH^0_{i+j-1})$$

$$q_1 = Pr(H^0_{i+j-2}/H^0_iH^1_{i+j-1}) \qquad q_5 = Pr(H^0_{i+j-2}/H^1_iH^1_{i+j-1})$$

$$q_2 = Pr(H^1_{i+j-2}/H^0_iH^0_{i+j-1}) \qquad q_6 = Pr(H^1_{i+j-2}/H^1_iH^0_{i+j-1})$$

$$q_3 = Pr(H^1_{i+j-2}/H^0_iH^1_{i+j-1}) \qquad q_7 = Pr(H^1_{i+j-2}/H^1_iH^1_{i+j-1})$$

$$\ldots(3.114)$$

We denote by :-

$\underline{QB_i}(j)$ the conditional density function vector on the LHS of equation (3.10a).

$\pi B_i(j)$ the transition matrix in equation (3.10a).

$\pi B_{m,i}(j)$ and $\pi B_{a,i}(j)$ the two matrices into which $\pi B_i(j)$ is split i.e.,

$$\pi B_i(j) = \pi B_{m,i}(j) . \pi B_{a,i}(j)$$

$\underline{PB_i}(j)$ the conditional density vector on the LHS of eqn. (3.11a).

and by ∂B the transition matrix in equation (3.12a).

We now describe the s equence of operations performed by the sequential detector in order to detect a fresh message symbol x_i .

Step 1.

Using exponentiators, the sequential detector evaluates f_o, f_1, ..., f_c according to eqn. (3.13) and putting $j = 1$. Along with these factors the factor q_o to q_7 are also evaluated.

Step 2.

At the beginning of the recursive proceedure in detecting x_i , the vector $\underline{QB_i}(0)$ has to be constructed

given the vector $\underline{QB}_{i-1}(1)$ evaluated during the detection of the previous symbol x_{i-1}. However, we note that :-

$$\underline{QB}_i(0) = \begin{bmatrix} p(\underline{r}_1,\ldots,\underline{r}_i/H_i^0\ H_{i-1}^0) \\ 0 \\ p(\underline{r}_1,\ldots,\underline{r}_i/H_i^0\ H_{i-1}^1) \\ 0 \\ p(\underline{r}_1,\ldots,\underline{r}_i/H_i^1\ H_{i-1}^0) \\ 0 \\ p(\underline{r}_1,\ldots,\underline{r}_i/H_i^1 H_{i-1}^1) \\ 0 \end{bmatrix}, \quad \underline{QB}_{i-1}(1) = \begin{bmatrix} p(\underline{r}_1,\ldots,\underline{r}_i/H_{i-1}^0 H_i^0) \\ p(\underline{r}_1,\ldots,\underline{r}_i/H_{i-1}^0 H_i^1) \\ 0 \\ 0 \\ 0 \\ 0 \\ p(\underline{r}_1,\ldots,\underline{r}_i/H_{i-1}^1 H_i^0) \\ p(\underline{r}_1,\ldots,\underline{r}_i/H_{i-1}^1\ H_i^1) \end{bmatrix}$$

and hence a mere transfer of components achieves the desired result. The sequential detector then goes on to multiply the vector $\underline{QB}_i(0)$ by the matrices $\pi B_{i,a}(j)$ and $\pi B_{i,m}(j)$ which process involves 2^{L+2} additions (8 in this case) and multiplications. With this, the vector $\underline{QB}_i(1)$ has been evaluated.

Step 3 .

From the vector $\underline{QB}_i(1)$ and using eqn. (3.12a), the detector computes the elements of $\underline{PB}_i(j)$ which process involves two additions.

Step 4 .

The probability ratio is then computed from the elements of $\underline{PB}_i(j)$. If $j = N_T$, the ratio is tested against unity and depending on whether it is lesser or greater than unity a decision regarding x_i is taken.

Step 5.

The following step is performed only when $j = 1$. The elements of the vector $\underline{QB}_i(1)$ are stored as they will be needed in the detection of the next symbol x_{i+1}.

Steps 1 to 4 are then repeated for $j = 2,3,\ldots,N_T$ until at $j = N_T$, the test is terminated at step 4 with a decision on the symbol x_i. Thus we see that the sequential detector employs $(2^{L+2} + 2)$ adders,

2^{L+2} multipliers,

and storage sufficient for $(2^{L+2} + 2^{L+1})$ analog samples constituting the elements of the vectors $\underline{QB}_i(j)$ and $\underline{QB}_i(1)$ respectively. This is as opposed to the $(2^{D+1} + 2)$ adders,

2^{D+1} multipliers,

and storage space for 2^D analog samples required by the fixed decoding delay receiver structure suggested by Abend and Fritchman.

Hence, we see that this receiver which is akin to the FSS detector considered in the example of target detection

in Radar because it examines sequentially a fixed number D of future symbols before coming to a decision, has a reduced computational complexity as compared to the recursive structure considered in Section 1. However, the decoder speed is clearly reduced by a factor of $\frac{1}{D+1}$ as a result. To make up for this loss, we introduce the sequential detector which retains the computational advantages of the FSS detector.

3.4 The Sequential Detector

The proceedure adopted by a sequential detector has already been detailed in Chapter II and we will therefore confine ourselves to specifying the specific sequential detection test (SDT) adopted by this detector. The sequential detector begins with the vector $\underline{QB}_{i-1}$ (1) of conditional density functions evaluated during the previous detection process. Examining an additional future received symbol each time, the receiver evaluates successively the probability ratio : -

$$\frac{p(\underline{r}_1,\ldots,\underline{r}_{i+j}/H_i^0)}{p(\underline{r}_1,\ldots,\underline{r}_{i+j}/H_i^1)} \qquad \text{for } j = 1, 2, \ldots, D$$

using exactly the same proceedure as that of the FSS detector

outlined in 3.3 and tests the ratio against a predetermined threshold A.

If $\dfrac{p(\underline{r}_1,\ldots,\underline{r}_{i+j}/H_i^0)}{p(\underline{r}_1,\ldots,\underline{r}_{i+j}/H_i^1)} \geqslant A$ the detector declares $x_i = 0$

If $\dfrac{p(\underline{r}_1,\ldots,\underline{r}_{i+j}/H_i^0)}{p(\underline{r}_1,\ldots,\underline{r}_{i+j}/H_i^1)} \leq \dfrac{1}{A}$ the detector declares $x_i = 1$

and if neither is true, the detector examines the next future symbol and recomputes the probability ratio for $j = j + 1$. If for all j $0 < j < D$, the test does not result in a decision, the detector tests the ratio

$\dfrac{p(\underline{r}_1,\ldots,\underline{r}_{i+D}/H_i^0)}{p(\underline{r}_1,\ldots,\underline{r}_{i+D}/H_i^1)}$ against a unity threshold thereby

ensuring forcible termination of the test. This SDT is commonly referred to as the sequential probability ratio test (SPRT).

All the statements made in Chapter II regarding the general sequential detector hold good for this detector also. The average sample number determines the maximum bit rate capability of this detector and the variable decoding delay causes a buffer requirement both at the input and

output of the detector. Fig. 3.1 shows how such a sequential detector may be implemented. The input section computes the conditional density functions f_o, f_1, f_2, f_3.... using exponentiators. Storage section A stores the vector of conditional density functions $\underline{QB}_{i-1}(1)$ needed to begin the detection of the symbol x_i. The multiply and add section performs the additions and multiplications indicated by the transition matrix $\pi B_i(j)$. in eqn. (3.10a) and the adders perform the addition indicated by eqn. (3.12a) in order to obtain the components of vector $\underline{PB}_i(j)$. The main storage section stores the components of the vector $\underline{QB}_i(j)$ needed during a detection process and finally, the threshold device checks the probability ratios appearing at its input against the threshold A. An input and output buffer are of course needed on account of the varying decoding delay.

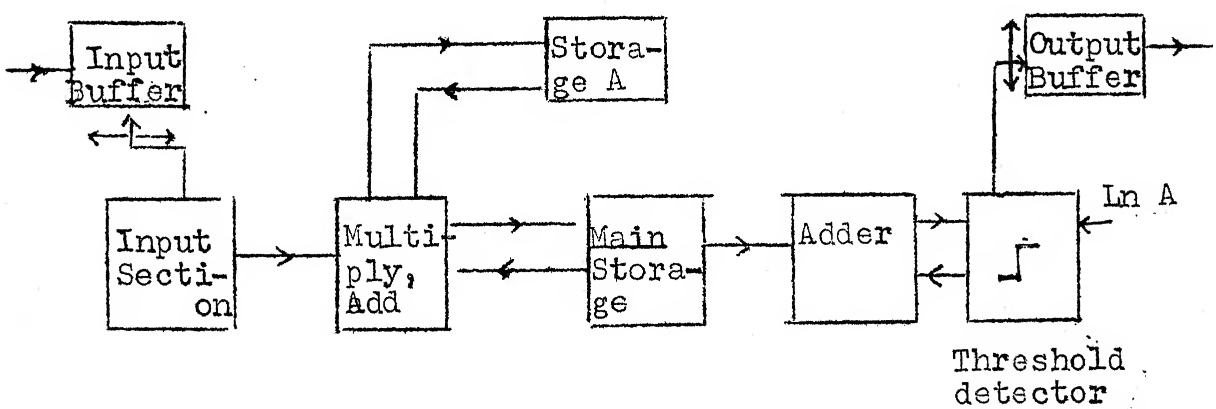


Fig. 3.1. Implementation of the sequential detector employing the SPRT.

3.5 Bound on the Probability of Error of the Sequential Detector.

We will now proceed to bound the error probability of the sequential detector using the SPRT. We will designate by PE (j) the probability of making an error in detection after j future received samples $\underline{r}_i$, $\underline{r}_{i+1}$, $\underline{r}_{i+2}, \ldots, \underline{r}_{i+j-1}$ have been observed. At this stage, the observation space is divided into three regions :-

R_o where $R_i(j) \geqslant A$ and x_i is declared $= 0$,

R_1 where $R_i(j) \leq \frac{1}{A}$ and x_i is declared $= 1$,

R_c where $\frac{1}{A} < R_i(j) < A$ and no decision is taken. Here $R_i(j)$ refers to the ratio :-

$$\frac{p(\underline{r}_1, \underline{r}_2, \ldots, \underline{r}_{i+j}/H_i^0)}{p(\underline{r}_1, \underline{r}_2, \ldots, \underline{r}_{i+j}/H_i^1)} .$$

Hence over R_o we have :-

$$p(\underline{r}_1, \underline{r}_2, \ldots, \underline{r}_{i+j}/H_i^0) \geqslant A \;\; p(\underline{r}_1, \underline{r}_2, \ldots, \underline{r}_{i+j}/H_i^1)$$

Integrating over the entire observation space, we obtain:-

$$\Pr(\text{ say } H_i^0/H_i^0) \geqslant A \ \Pr(\text{ say } H_i^1 / H_i^0) \qquad \ldots (3.15a)$$

Similarly, we can obtain by considering the region R_1 that:-

$$\Pr(\text{ say } H_i^1/H_i^1) \geqslant A \ \Pr(\text{ say } H_i^0 /H_i^1) \qquad \ldots (3.15b)$$

Adding the left and right hand sides of equation (3.15a) and (3.15b), we obtain :-

$$\Pr(\text{ correct decision }) \geqslant A \Pr(\text{ erroneous decision})$$

i.e. $$(1 - PE(j)) \geqslant A \ PE(j)$$

$\therefore$ we have $$PE(j) \leq \frac{1}{1+A} \qquad \ldots (3.16)$$

Clearly, the equation (3.16) holds for all j $0 < j < D$.

The bit error probability of the sequential detector is given by :-

$$PE = \sum_{j=0}^{D-1} p(j).PE(j) + p(A,D).PE(A,D)$$

$$\leq \frac{1}{1+A} \sum_{j=0}^{D-1} p(j) + p(A,D).PE(A,D).$$

and hence $$PE \leq \frac{1}{1+A} + p(A,D).PE(A,D). \qquad \ldots (3.17)$$

Therefore the bit error probability of the sequential detector can be made close to that of the optimal fixed delay detector(whose probability of error is $\geq p(A,D) \cdot PE(A,D)$) by chosing a suitably large value of A. We cannot choose an arbitrarily high value of A since this would increase the ASN thus decreasing the decoding speed of the detector.

3.6 Conclusions.

The table below shows the number of multipliers, adders and the storage space needed by the optimum fixed decoding delay receiver using the structure suggested by Abend and Fritchman as well as that needed by the sequential detector.

Table 3.1

	Fixed delay decoder	Sequential detector
Adders	$2^{D+1} + 2$	$2^{L+2} + 2$
Multipliers	2^{D+1}	2^{L+2}
Storage space	2^{D}	$2^{L+2} + 2^{L+1}$

Hence, we see that the amount of saving that will result will depend on the relative values of D and L. If one considers $D = 2(L + 1)$ as suggested by the results presented by Abend and Fritchman (AB - 70) for the case of intersymbol

interference, then for a typical value of L = 5 we retabulate the requirement of computational elements.

	Fixed Delay Decoder	Seq. Detector
Adders	$2^{11} + 2 = 2050$	$2^7 + 2 = 130$
Multipliers	$2^{11} = 2048$	$2^7 = 128$
Storage	$2^{10} = 1024$	$2^7 + 2^6 = 192$

Table 3.2

It might be argued that the sequential detector has a loss in decoding speed by a factor of $\frac{1}{\text{ASN}}$ and hence it is only fair to compare the sequential detector with a fixed delay decoder using a sequential implementation having $\frac{1}{\text{ASN}}$ times the number of elements shown in Table 3.1. However, even if one considers a value of ASN = (L + 1) we see that the difference in requirements is still considerable and what is more will grow exponentially with the value of L.

Thus, the sequential detector employs a recursive structure which for moderate to large values of L and D will result in a considerable saving of computational and storage elements. However, both detectors considered above need exponentiators for implementation and, considering the present

state of hardware development, it seems unlikely that such a receiver will be implemented in practice. We hence turn our attention to a more practical decoding algorithm in the next chapter namely, the maximum likelihood (ML) sequence detection algorithm.

CHAPTER IV

SEQUENTIAL DETECTION EMPLOYING A ML CRITERION

In this Chapter, we consider a sequential detector which uses a maximum likelihood criterion for conducting the sequential detection test (SDT).

The Viterbi algorithm receiver which is a practical version of the maximum likelihood (ML) sequence estimator employs a recursive structure having an excessive storage requirement (Viterbi [VIT-71]). We therefore propose in Section 1, an alternative recursive structure which does away with this storage requirement though at the cost of a serious loss in decoding speed. We then present in Section2, a modification of the above sequential detector which while retaining the advantage of a greatly reduced storage requirement makes up to a large extent the loss in decoding speed. We show in Section 3 by obtaining an upper bound on the bit error probability of this detector that its error performance is comparable to that of the Viterbi algorithm detector. In Section 4, an upper bound on the Average Sample Number (ASN) of the sequential detector is presented. Here we also consider its buffer requirements before going on to conclude the Chapter in Section 5 with a presentation of simulation results and the conclusions to be drawn therefrom.

4.1 The non-recursive ML Sequence Estimator.

As we have seen in the previous chapter, both the optimum detector and the sequential detector using the SPRT have need of exponentiators which, in practice, are difficult to realize. It is for this reason that the search for suboptional detectors combining good error performance with ready implementability continues. One such decoder is the ML sequence decoder. The ML sequence detector is that which examines all possible message sequences before choosing the one which is the most likely given the received symbols. If a long stream of bits is transmitted as in practice, then it will not be practical to store and compare all possible transmitted sequences and accordingly, a modified version of ML sequence detection namely the Viterbi detection algorithm is often used.

The Viterbi algorithm receiver described in [VIT-68], [VIT-71], [FOR-71] suffers from the disadvantage of having a storage and computational complexity which grow exponentially with the memory of the FSM. For example, practical decoders for rate (k/n) convolutional codes having a constraint length K and employing a decoding delay of 4K have a storage requirement of $4kK.\,2^{k(K-1)}$ bits. This storage requirement arises from the nature of the algorithm which recursively evaluates the "survivors" as well as their

"metrics". An alternative procedure would be to evaluate the survivor metrics afresh for each new bit being decoded thus doing away with the need to store the survivors themselves. To see how this may be done, consider the trellis drawn below for the case of a rate $\frac{1}{2}$ convolutional code having a constraint length $K = 3$ and whose generator polynomials are given alongside the figure.

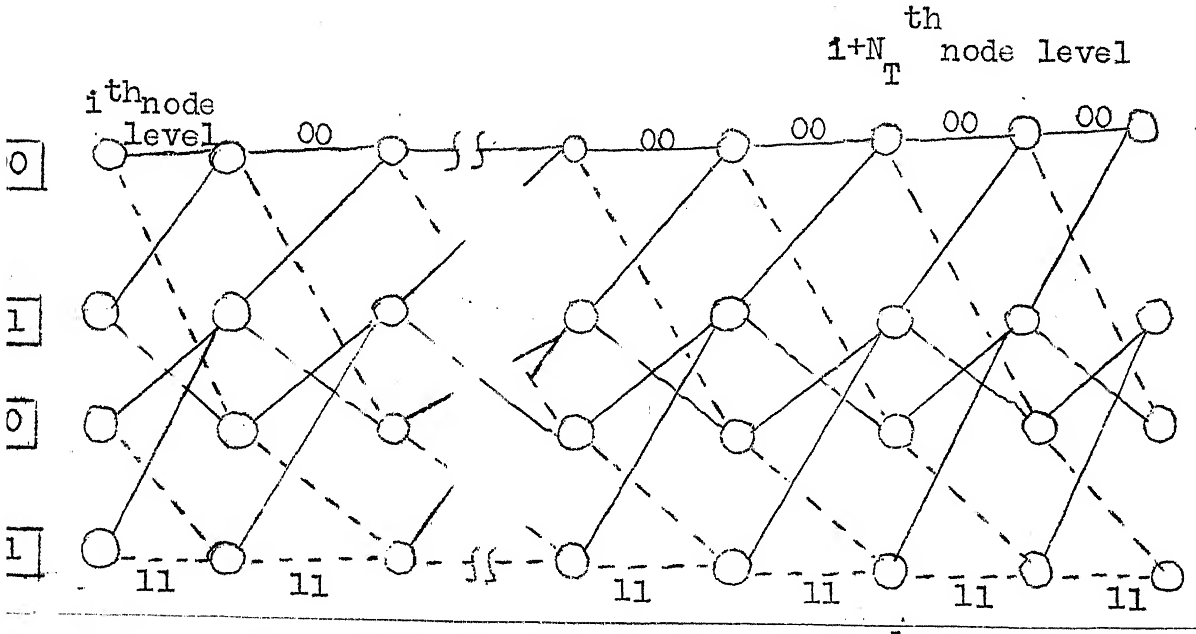


Fig.4. Trellis diagram for a rate $\frac{1}{2}$ convolutional code with Generator Polynomials

$$\left.\begin{aligned} G_1(D) &= 1 + D + D^2 \\ G_2(D) &= 1 + D^2. \end{aligned}\right\}$$

In the Viterbi algorithm, one moves in the trellis from one node level to the next recursively evaluating the metrics of the survivors as well as the survivors themselves

at each node level. In this recursive proceedure, the i^{th} message symbol of each survivor is computed during the recursive computation involved in going from the i^{th} to the $(i + 1)^{th}$ node levels. However, a decision regarding the i^{th} message symbol is reached only after the survivors as well as the survivor metrics at the $(i + N_T)^{th}$ node level have been computed. Clearly therefore, it is necessary to store the i^{th} message symbol while moving from the $(i+1)^{th}$ to the $(i + N_T)^{th}$ node level of the trellis. Similarly, it is necessary to store the $(i+1)^{th}$ message symbol during each of the recursive computations needed in order to move from the $(i+2)^{th}$ node level to the $(i + N_T + 1)^{th}$ level and so on..... . Hence we need to store the $(i+1)^{th}$ to the $(i + N_T)^{th}$ message symbols of each survivor at the $(i + N_T)^{th}$ node level making a total of $N_T \cdot N_F$ bits in all. (The i^{th} symbol may be discarded as soon as a decision is made). This is the reason for the survivor storage requirement of the Viterbi decoder.

We had earlier on mentioned a recursive proceedure in which the need for this storage requirement is done away with. In this proceedure, we propose that we start with the metrics of the survivors at the ith node level at the start of the decision making process in volving the i^{th} message symbol x_i instead of the metrics of those at the $(i+ N_T-1)^{th}$ node level.

The metrics of the survivors at the $(i+1)^{th}$, $(i+2)^{th}$,..., $(i+N_T)^{th}$ node levels are then successively evaluated using a set of add compare select operations (ACS) just as in the case of the Viterbi detector. During the recursive computation involved in moving from the i^{th} to the $(i+1)^{th}$ node levels, we compute the i^{th} message symbol of each of the survivors, but thereafter when we move on to the successive node levels, we do not concern ourselves with the successive message symbols of the survivors but maintain only the metrics of the survivors and the i^{th} message symbol corresponding to each survivor. Also, after reaching the $(i+1)^{th}$ node level, we store the metrics of the survivors at this node level since we will be in need of them to begin the detection of the next symbol x_{i+1}. After we have reached the $(i + N_T)^{th}$ node level we make the decision regarding the i^{th} message symbol just as in the case of the Viterbi detector since the i^{th} message symbol (and only the i^{th} message symbol) is available to us at this node level. Hence we have succeeded in reducing the storage requirement from $N_T . N_F$ bits to a mere N_F bits. (An additional storage space is needed by this detector however, to store the metrics of the $(i+1)^{th}$ node level needed to begin the detection of the symbol x_{i+1}). The gain in storage requirement is

achieved at the cost of a loss in decoding speed since we have to go through the recursive proceedure involved in going from one node level to the next N_T times.

Since the decoding speed of this receiver is clearly $\frac{1}{N_T}$ th that of the Viterbi decoder, we follow the steps taken in Chapter III under similar conditions, and introduce sequential detection to make up for this loss in decoding speed.

4.2. Sequential Detection Using a ML Criterion.

Before we discuss the SDT employed by this sequential detector, we introduce the following notation :-

We shall denote by S^o_{i+j} and S^1_{i+j}, the survivors at the $(i+j)^{th}$ node level having greatest metric amongst all those having $x_i = 0$ and $x_i = 1$ respectively.

The proceedure adopted by the sequential detector is identical to that mentioned in Chapter 1 and the recursive structure it employs is identical to that in the previous section. The sequential detector begins with the metrics of the survivors at the i^{th} node level and thereafter recursively evaluates using a set of ACS (add, compare, select) operations (see Heller and Jacobs [HJ- 71]),

the node metrics at successive node levels. After the metrics of the survivors at the $(i+j)^{th}$ node level have been evaluated, they are used to compute the ratio :-

$$\frac{p(\underline{r}_1, \underline{r}_2, \ldots, \underline{r}_{i+j} \,/\, S^0_{i+j} \text{ is the message sequence})}{p(\underline{r}_1, \underline{r}_2, \ldots, \underline{r}_{i+j} \,/\, S^1_{i+j} \text{ is the message sequence})}$$

which we shall denote by $R_i(j)$.

The sequential detection test is then carried out as follows:-

If $R_i(j) \geq A_j$ the receiver declares $x_i = 0$

If $R_i(j) \leq \frac{1}{A_j}$ the receiver declares $x_i = 1$

and if neither holds, the ratio $R_i(j+1)$ is evaluated using an additional received symbol $\underline{r}_{i+j+1}$ and the SDT applied to it and so on. The thresholds A_j are predetermined and will be shown to be related to the bit error probability of the receiver. Setting $A_{N_T} = 1$ ensures termination of the SDT at the $(i+N_T)^{th}$ node level in the event that a decision does not materialize earlier.

Like any other sequential detector, this detector is characterized by an ASN and by an input and output buffer requirement arising from the variable decoding delay involved. We deal with each of these topics seperately in the sections to follow.

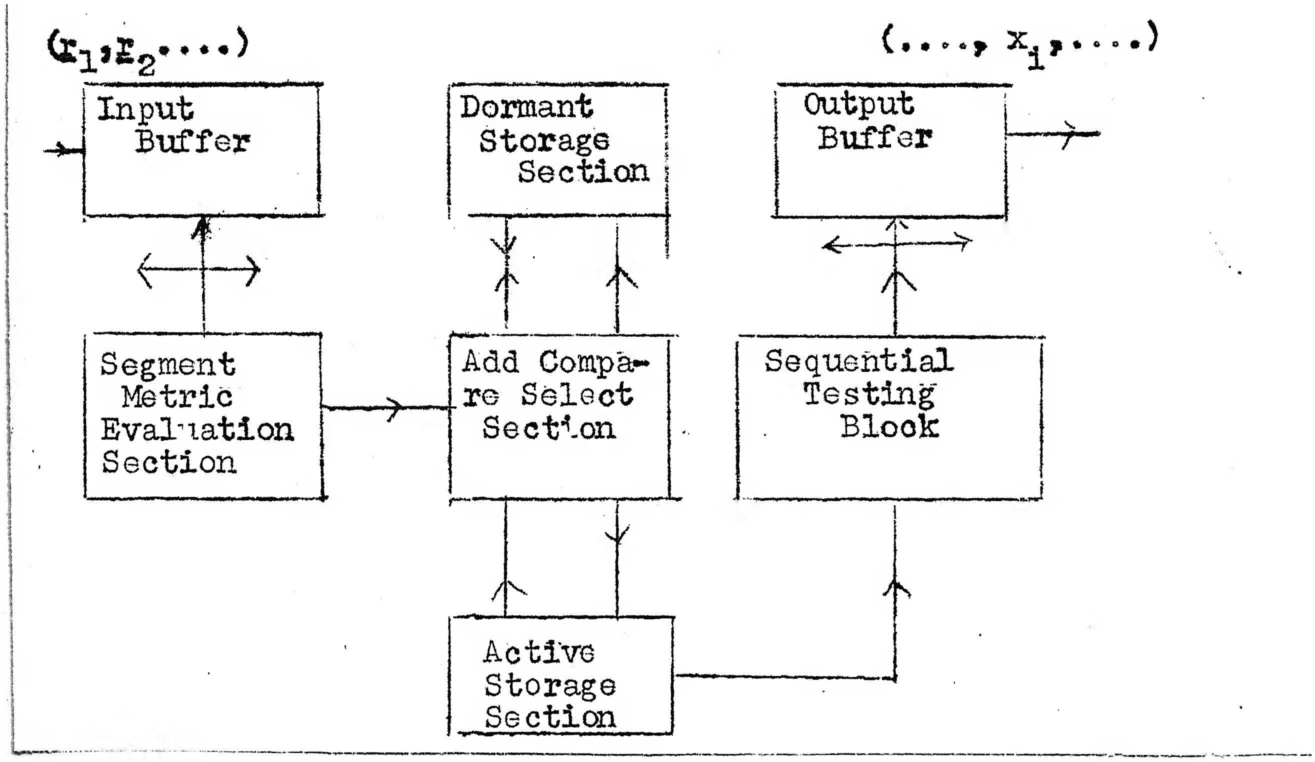


Fig. 4.1. Implementation of the Sequential Detector Employing a ML Criterion.

The sequential detector may be implemented as shown in the block diagram of Fig. (4.1). The input and output buffers are needed on account of the varying decoding delay. The dormant storage section stores the metrics of the survivors at the $(i+1)^{th}$ node level needed to begin the detection of the next symbol x_{i+1} whilst the bit x_i is being detected. The active storage section stores the metrics of the survivors at whatever node level the detector happens to be at during the detection of the present symbol x_i. The i^{th} message symbol of each of the survivors is also stored here. The sequential testing block uses the metrics of the survivors at the current node level in order to

determine whether a decision can be taken or not.

With this, our description of the sequential detection algorithm and its implementation is complete and we now go on to obtain an upper bound to the bit error probability of the sequential detector.

4.3 Bound on Bit Error Probability

4.3.1 The method employed to evaluate a bound on the bit error probability of the sequential detector is basically an extension of the generating function technique introduced by Viterbi to general FSM's (ref. Viterbi [VIT -71]). The technique basically consists of ennumerating exhaustively, all possible pairs of output sequences of the FSM. Associated with every ordered pair of output sequences, there is the probability that the 1st is the true output sequence and the second is the estimated output given that the 1st is the true output sequence. Now clearly, the sum of the probabilities associated with every ordered pair of output sequences in which the two sequences comprising a pair differ in their i^{th} symbol, will constitute a union bound on the probability of bit error of the sequential detector.

For the particular case of convolutional codes, a simplification is possible since convolutional codes are

I.I.T. KANPUR
CENTRAL LIBRARY
Acc. No. A 59510

group codes and hence the error probability is the same for every possible output sequence of the convolutional encoder. It is thus for this reason that the bound on error probability is evaluated considering the all zero sequence to be the true transmitted sequence.

To facilitate understanding of our derivation of an upper bound on the bit error probability of the sequential detector we now introduce some terminology.

4.3.2

The definitions to follow relate to the information transmission system considered in Chapter I and the block diagram has been reproduced here for the sake of convenience.

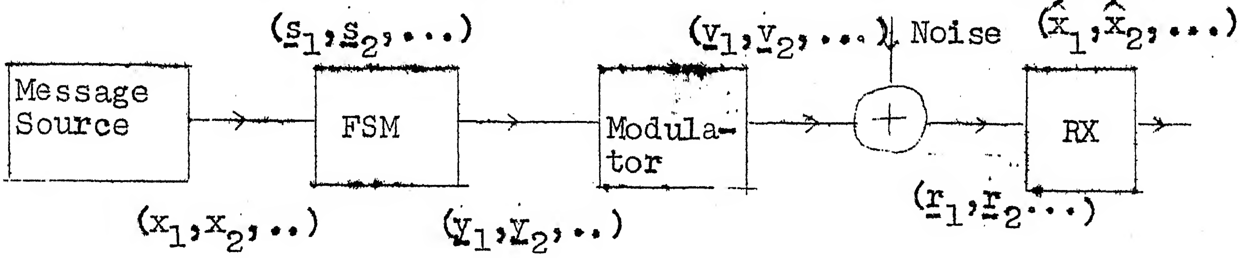


Fig. 4.2 Information Transmission System Model.

We will denote the :

estimated message sequence by $(\hat{x}_1, \hat{x}_2, \ldots.)$

estimated state sequence by $(\hat{\underline{s}}_1, \hat{\underline{s}}_2, \ldots\ldots)$

estimated output of the FSM to be the sequence

$(\hat{\underline{y}}_1, \hat{\underline{y}}_2, \ldots)$

and the estimate transmitted sequence by $(\hat{\underline{v}}_1, \hat{\underline{v}}_2, \ldots)$

We define :-

the sequence $\left[\binom{x_1}{\hat{x}_1}, \binom{x_2}{\hat{x}_2}, \ldots \right]$

to be the compared message sequence

$\left[\binom{\underline{s}_1}{\hat{\underline{s}}_1}, \binom{\underline{s}_2}{\hat{\underline{s}}_2} \ldots \right]$

to be the compared state sequence

$\left[\binom{\underline{y}_1}{\hat{\underline{y}}_1}, \binom{\underline{y}_2}{\hat{\underline{y}}_2} \ldots \right]$

to be the compared output sequence of the FSM

$\left[(\underline{v}_1 - \hat{\underline{v}}_1), (\underline{v}_2 - \hat{\underline{v}}_2), \ldots \right]$ to be the error transmitted sequence or simply, the error sequence.

$\underline{X}_i$ to be the 1st i-length sequent $[x_1, x_2, \ldots, x_i]$ of the message sequence,

$\underline{N}_F$ to be the number of states of the FSM,

<u>the weight function associated with a compared message sequence symbol</u> $\left[\begin{matrix} x_i \\ \hat{x}_i \end{matrix} \right]$ and a compared state $\left[\begin{matrix} \underline{s}_i \\ \hat{\underline{s}}_i \end{matrix} \right]$ to be the function $\frac{1}{D}.D^{w^2}$ where 'w' is the Euclidean weight of the corresponding error vector $[\underline{v}_i - \hat{\underline{v}}_i]$,

$$w = [(\underline{v}_i - \hat{\underline{v}}_i)^T (\underline{v}_i - \hat{\underline{v}}_i)]^{1/2}$$

and the symbol 'D' is an indeterminate.

The <u>weight function associated with a compared message sequence</u> to be the product of the weight functions associated with the individual compared message sequence and state sequence symbols. The weight function of the sequence

$\left(\binom{x_1}{\hat{x}_1}, \binom{x_2}{\hat{x}_2} \ldots \right)$ enables us to evaluate the probability that the sequence $x_1, x_2 \ldots$ was transmitted but was estimated as the sequence $\hat{x}_1, \hat{x}_2, \ldots$.

The <u>compared state and compared trellis diagrams</u> in which each vertex corresponds to a compared state $\begin{bmatrix} \underline{s}_i \\ \hat{\underline{s}}_i \end{bmatrix}$ and each transition between two vertices is accompanied by the weight function associated with the transition.

A <u>null state</u> to be a state in either the compared state of the compared trellis diagram in which $(\underline{s}_i \equiv \hat{\underline{s}}_i)$.

N_c to be the number of states in the compared state diagram. $N_c = N_F^2$ in general.

A <u>merging sequence</u> to be a compared message sequence in the compared trellis diagram which begins and ends on a null state without passing through an intermediate null state.

A <u>branching sequence</u> to be a compared message sequence in the compared trellis diagram which begins on a null state

and which thereafter does not pass through a null state.

The <u>weight function of a node</u> in the compared trellis diagram to be the sum of the weight functions of all sequences leading to it which have branched off from the true message sequence at an earlier mode level.

The vector $G_j]$ to be a vector having N_c components whose elements are the weight functions of the nodes at the j^{th} node level of the compared trellis diagram. The 1st N_F elements of $G_j]$ give the weight functions of the null states at the j^{th} node level, and finally, <u>the weight function matrix</u> $[W]$ to be a $(N_c \times N_c)$ square matrix whose $(1,n)^{th}$ element gives the weight function of the transition from the 1^{th} node at a node level in the compared trellis diagram to the m^{th} node at the very next node level. Clearly, we have

$$G_{j+1}] = [W] \cdot G_j] \qquad \ldots(4.1)$$

To illustrate the above, we consider an example of a FSM and present the corresponding trellis and state diagrams. The FSM is the one considered earlier as a model for a channel with intersymbol interference. (see Fig. 4.3). We note here that for the particular case of intersymbol interference, the null states of the compared

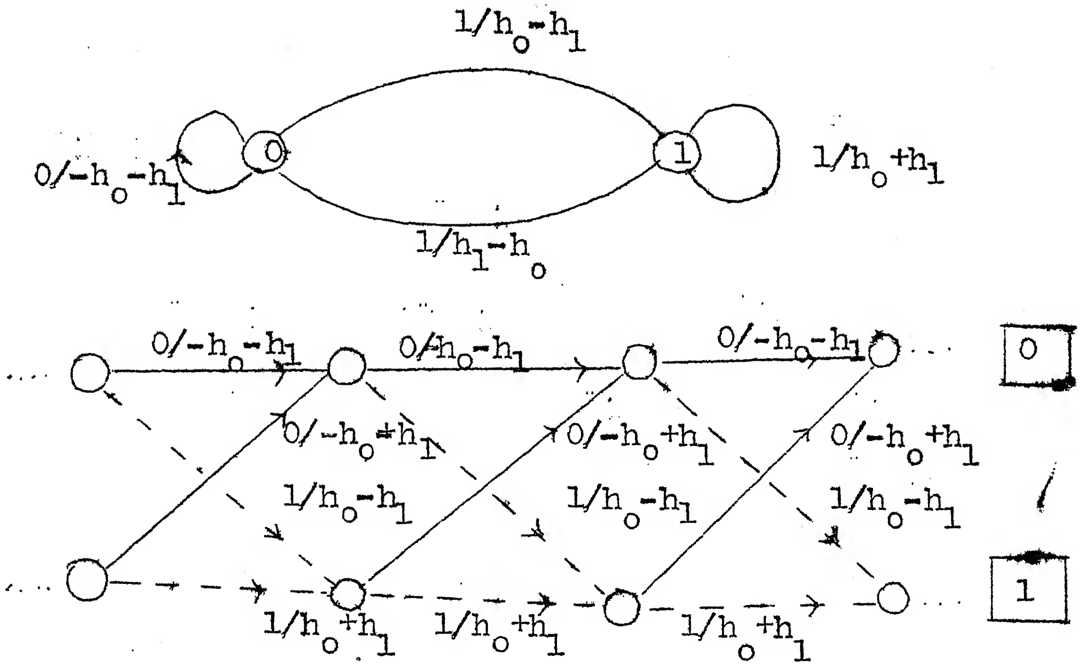


Fig. 4.3a State and Trellis Diagrams of the FSM.

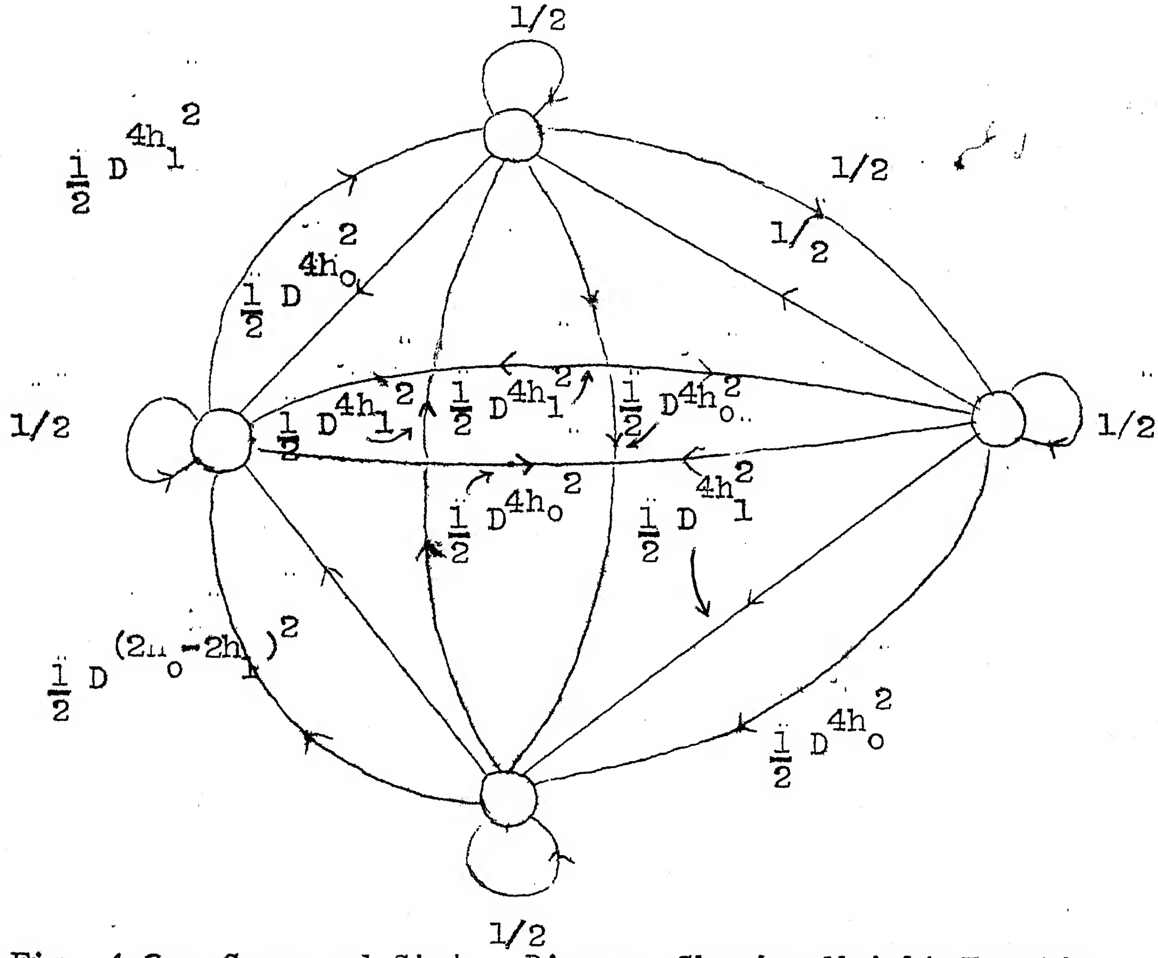


Fig. 4.3c. Compared State Diagram Showing Weight Functions Associated with Each Transition.

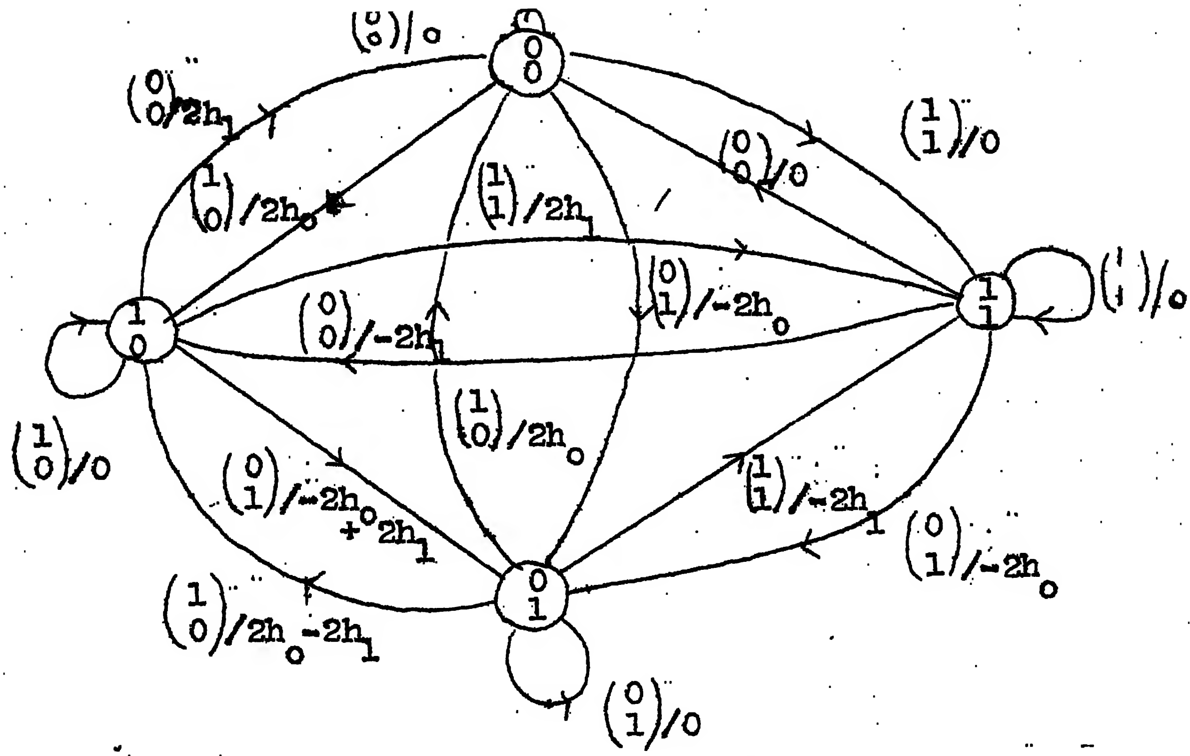


Fig. 4.3b Compared State Diagram Showing Input and Output in each Transition.

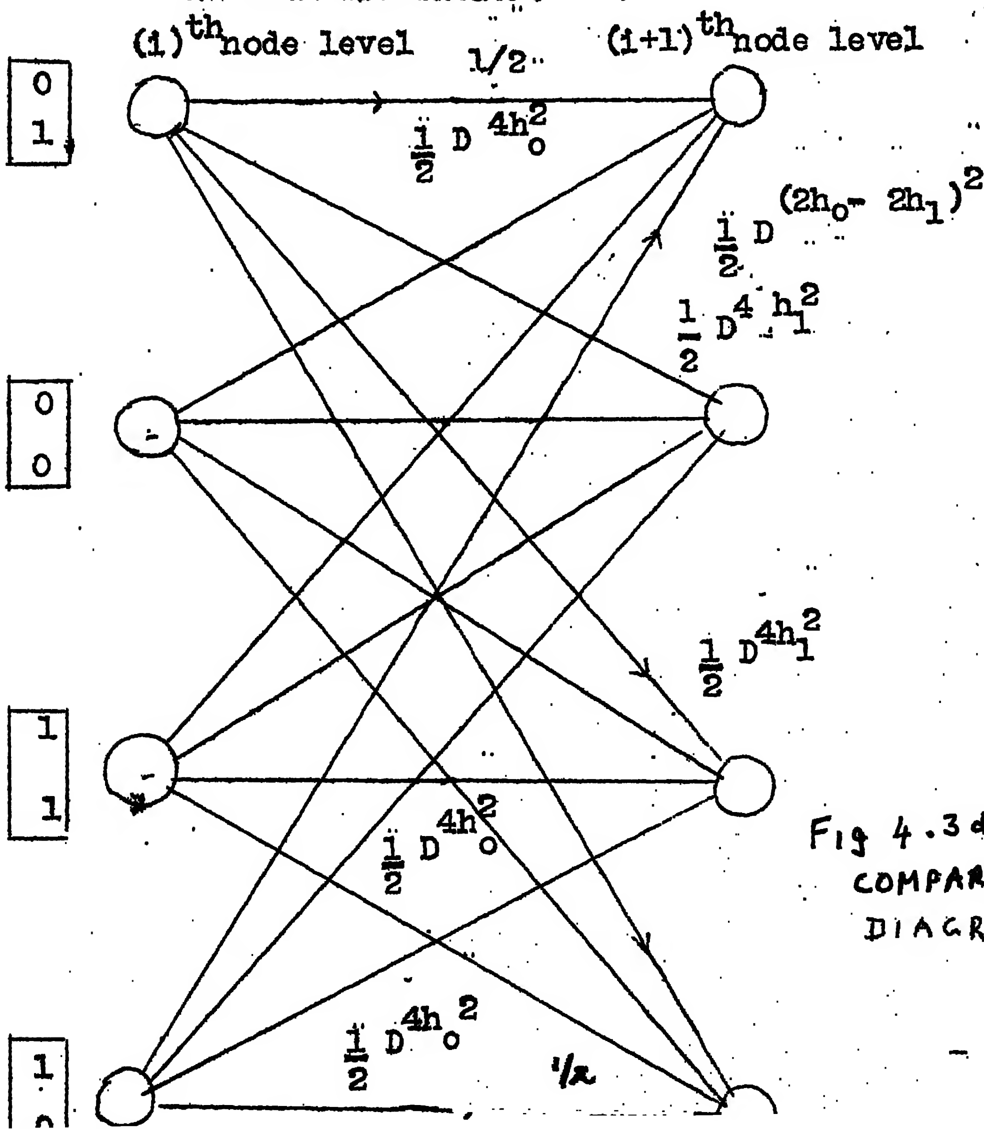


Fig 4.3d COMPARED TRELLIS DIAGRAM

state diagram are equivalent and can be replaced by a single state if state minimization is desired. This is because the FSM is linear i.e., the equations δ and w are linear equations. However, since this is not true of general FSM's, we do not attempt to simplify the compared state diagram here.

4.3.3.

Having completed the necessary preliminaries, we now proceed to derive the bound on the error probability of the sequential detector. Consider the detection of the symbol x_i in which the detector recursively evaluates the ratio

$$\frac{p(\underline{r}_1,\underline{r}_2,\ldots,\underline{r}_{i+j}\,/S^0_{i+j} = \text{message sequence})}{p(\underline{r}_1,\underline{r}_2,\ldots,\underline{r}_{i+j}/S^1_{i+j} = \text{message sequence})}$$

for $j = 1,2,\ldots$

in succession. If the true message sequence remains a survivor, then an error can only take place if a message sequence having a symbol other than x_i as its i^{th} symbol, has at the $i+j^{th}$ node level of the trellis diagram, a metric exceeding that of the true message sequence by a factor $\ln A_j$. If the actual message sequence fails to remain a survivor, then also an error may result.

From this we conclude that an error in detecting the symbol x_i can occur only if at least one of the events listed below takes place. Though we do not mention so in each case, the sequences considered in each of these three events are those which differ in their i^{th} symbol from the symbol x_i.

Event 1.

A merging sequence branching out from a node level l $(l \leq i)$ and merging at node level m with the actual message sequence $(i \leq m \leq i + N_T)$ has a metric exceeding that of the latter at the m^{th} node level.

A compared message sequence corresponding to such a sequence is shown labelled E1 in Fig. 4.3. which shows the error trellis diagram for the FSM considered in Section 4.3.2. We note that such a merging sequence would cause an error in the Viterbi decoder also and indeed, in the ideal ML decoder having an infinite decoding delay, this event is solely responsible for error in decision making.

Event 2.

A branching sequence branching out from a node level $l(l \leq i)$ has at the $(i + N_T)^{th}$ node level a metric exceeding that of the true message sequence.

If $(\underline{y}_1, \underline{y}_2, \ldots, \underline{y}_i)$ is the transmitted sequence corresponding to the sequence $\underline{X}_1^1$, we have that the probability of the event is given by :

$$\Pr\left[\frac{e^{-\frac{1}{2\sigma_n^2}\sum_{j=1}^{i}(\underline{r}_j-\underline{y}_j)(\underline{r}_j-\underline{y}_j)}}{e^{-\frac{1}{2\sigma_n^2}\sum_{j=1}^{i}(\underline{r}_j-\underline{v}_j)(\underline{r}_i-\underline{v}_j)}} \quad A\right]$$

$$= \operatorname{erfc}\left[\frac{d}{\sigma_n} + \frac{2\sigma_n}{d}\ln A\right] \quad \ldots (4.3)$$

where σ_n = variance of the additive Gaussian white noise introduced by the channel.

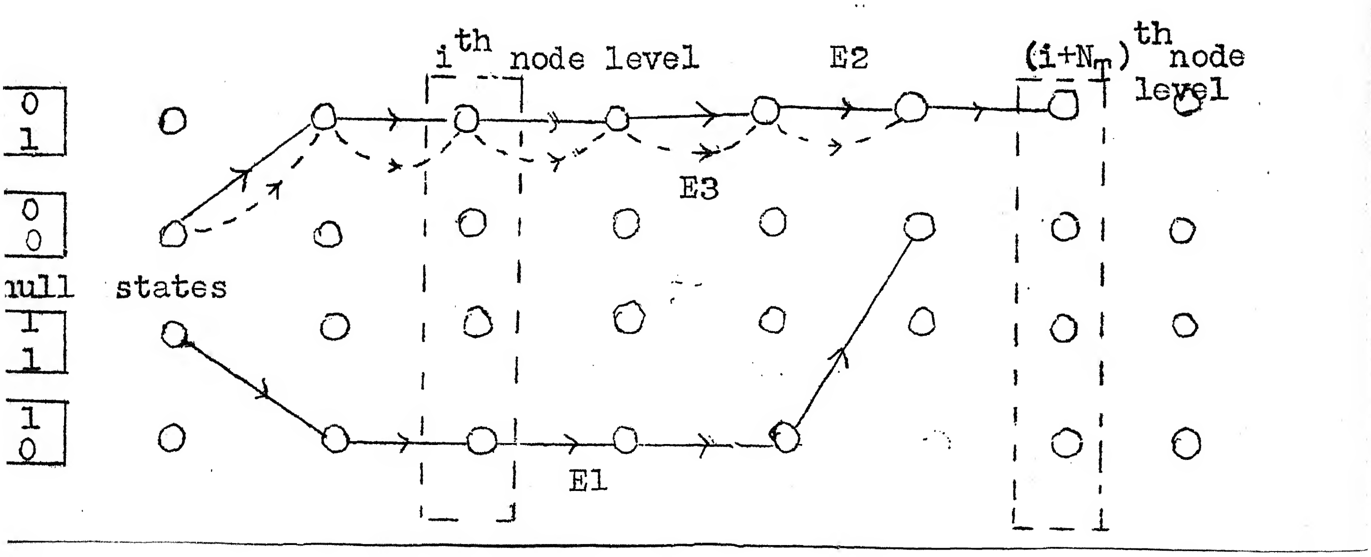


Fig. 4.4. Compared Trellis Showing Sequences Causing Error Events.

and where d is the Euclidean weight of the sequence $(\underline{\gamma}_1 - \underline{y}_1, \underline{\gamma}_2 - \underline{y}_2, \ldots)$ and is hence given by :-

$$d = \left[\sum_{j=1}^{i} (\underline{\gamma}_i - \underline{y}_i)^T (\underline{\gamma}_i - \underline{y}_i) \right]^{1/2} \qquad \ldots(4.4)$$

By invoking the union bound, one may upper bound the probability of error in decoding the symbol x_i by summing the corresponding probabilities for every message sequence of the type listed in events 1 to 3 above. However, since corresponding to each such message sequence and an actual message sequence we have a compared message sequence whose weight function is known, we sum instead the weight functions of the corresponding compared message sequences. It might appear from a consideration of events 1 to 3 that one might have to consider an infinite number of sequences but, in practice this is not so since branching or merging sequences which branch away from node levels from the i^{th} node level are at great distances from the actual message sequence and hence the corresponding probabilities negligible. We therefore confine our attention to those sequences branching off at or after the $(i - N_S)^{th}$ node level where we define the $(i - N_S)^{th}$ node level to be a node level sufficiently 'far' in the above sense of the word from the i^{th} node level. The summation of the weight functions mentioned above can now be carried out as follows:-

Starting at the $(i-N_S)^{th}$ node level of the compared trellis diagram, we assign a weight function of 1 to the null states and 0 to the others. We then arrive at the weight functions of the nodes at the i^{th} node level by successive multiplication of the weight function vector $G_{i-N_S}]$ by the weight function matrix $[W]$ as below:-

$$G_{i-N_S+j}] = [W]. \; G_{i-N_S+j-1}]$$

$$j = 1,2,\ldots, N_S \qquad ..(4.5)$$

At the i^{th} node level, we modify the weight function matrix W to exclude those segments corresponding to message symbols identical to x_i and obtain $G_{i+1}]$ through multiplication as before. To arrive at the weight functions of the nodes at the succeeding node levels, we modify the weight function $[W]$ and replace by 0 the weight functions of the segments branching out from the null states since we are not interested in considering sequences branching off the true message sequence path at node levels beyond the i^{th} node level as the event that they represent constitutes a sub event of an event already considered in 1.

As an example, consider a sequence that branches off at the $i-j^{th}$ node level but which passes through a null state at the $(i+k)^{th}$ node level. Then consider the event

that the sequence has at the node level $(i+k+1)$ a metric exceeding by $\ln A_i$ that of the true message sequence. Clearly, this can happen if and only if the merging sequence obtained by considering the section of this sequence in between the $(i-j)^{th}$ and $(i+k)^{th}$ node levels has a metric exceeding that of the true message sequence since otherwise this sequence would not remain a survivor at the $(i+k)^{th}$ node level. Hence our statement earlier on to the effect that this event was a subevent of an event whose contribution to the union bound on the probability of error had already been considered.

As the last step, we proceed to recursively evaluate $G_{i+j}]$ for $j = 2,3,\ldots N_T$ by multiplying this time by the modified weight function matrix. Having done this, it is a simple matter to bound the error probability as :-

$$PE \leq \sum_{i=1}^{N_T-1} \sum_{j=N_F+1}^{N_C} G_i(j) \Bigg|_{D^d = \operatorname{erfc}\left[\frac{\sqrt{d}}{\sigma_n} + \frac{\sigma_n \ln A_i}{2\sqrt{d}}\right]}$$

$$+ \sum_{j=N_F+1}^{N_C} G_{N_T}(j) \Bigg|_{D^d = \operatorname{erfc}\left[\frac{\sqrt{d}}{\sigma_n}\right]}$$

$$+ \sum_{i=1}^{N_T} \sum_{j=1}^{N_F} G_i(j) \Bigg|_{D^d = \operatorname{erfc}\left[\frac{\sqrt{d}}{\sigma_n}\right]} . \qquad \ldots(4.6)$$

The last two terms constitute a bound on the probability of bit error of the Viterbi algorithm detector though its derivation is slightly different from that by Viterbi in [VIT - 71]. By making the A_i's sufficiently large we can make the additional term small thereby consolidating the statement made earlier that the error performance of the sequential detector is comparable to that of the Viterbi algorithm detector.

4.4 The ASN and the Buffer Requirement of the Sequential Detector.

4.4.1 Bound on the ASN: We will first proceed to derive in this Section, an upper bound to the ASN of the sequential detector. We rewrite the expression for the Average Sample Number given in Eqn. (2.1) as follows :-

$$ASN = \sum_{j=1}^{N_T + 1} j \cdot p(j) \qquad \ldots(4.7)$$

where p(j) here denotes the probability that the SDT terminates the detection process of the symbol x_i at the $(i + j)^{th}$ node level of the trellis after observing the j^{th} future symbol $\underline{r}_{i+j-1}$.

The derivation of the bound on the ASN closely follows the derivation of the bound on the bit error probability of the sequential detector. To begin with we seek an upper bound on the probability that the test for the

message symbol x_i does not terminate at the $(i+j)^{th}$ node level given that $A_k = \infty \; \forall \; k$ s.t. $(0 \le k < j)$. We shall denote the probability of this event by $q(j)$. We draw a parallel between this event and the event representing a decision error at the $(i+j)^{th}$ node level. The latter event is an event of the type Event 3 listed in Section 4.3. An upper bound for the event was obtained in Section 4.3 by utilizing the union bound and summing the probabilities of every branching sequence of the type listed in Event 3 having a metric exceeding that of the true message sequence by an amount equal to $\ln A_j$. Similarly, we can obtain an upper bound on the probability of non termination of the sequential test at the $i+j^{th}$ node level by summing the probabilities of every message sequence of the type considered in Event 3 having a metric exceeding that of the true message sequence by an amount equal to $\ln \frac{1}{A_j}$. Hence we can upper bound $q(j)$ as follows:-

$$q(j) \le \sum_{k=N_F+1}^{N_c} G_{i+j}(k) \Big|_{D^d = \operatorname{erfc}\left[\frac{\sqrt{d}}{\sigma_n} - \frac{\sigma_n . \ln A_j}{2\sqrt{d}}\right]}$$

$$+ \sum_{k=1}^{N_F} G_{i+j}(k) \Big|_{D^d = \operatorname{erfc}\left[\frac{\sqrt{d}}{\sigma_n}\right]} \qquad (4.8)$$

Since the probability of non-termination at the $(i+j)^{th}$ node level will decrease if one uses instead, a finite set of values for $A_k \;\forall\; j,\; 0 \leq k \leq j$ we can write:-

$$1 - \sum_{k=1}^{j} p(k) \leq q(j)$$

Summing over all j, $1 \leq j \leq N_T$ on both sides, we obtain:=

$$N_T - \sum_{j=1}^{N_T} \sum_{k=1}^{j} p(k) \leq \sum_{j=1}^{N_T} q(j)$$

Adding one to both sides and noting that $\sum_{k=1}^{N_T+1} p(k) = 1$, we write,

$$(N_T+1) \cdot \left(\sum_{k=1}^{N_T+1} p(k) \right) - \sum_{j=1}^{N_T} \sum_{k=1}^{j} p(k) \leq \sum_{j=1}^{N_T} q(j) + 1$$

$$\therefore \sum_{k=1}^{N_T+1} p(k) \cdot k \leq 1 + \sum_{j=1}^{N_T} q(j)$$

$$\text{i.e. } ASN \leq 1 + \sum_{j=1}^{N_T} \sum_{k=N_F+1}^{N_c} G_{i+j}(k) \Bigg|_{D^d = \operatorname{erfc}\left[\frac{\sqrt{d}}{\sigma_n} - \frac{\sigma_n \ln A_j}{2\sqrt{d}}\right]}$$

$$+ \sum_{j=1}^{N_T} \sum_{k=1}^{N_F} G_{i+j}(k) \Bigg|_{D^d = \operatorname{erfc}\left[\frac{\sqrt{d}}{\sigma_n}\right]} \qquad \ldots (4.9)$$

Equation (4.9) gives the desired bound on the ASN. We note that the last term on the RHS of eqn. (4.9) can be neglected since it will be small compared to the first two terms.

4.4.2. Optimal Threshold Setting

It should be clear from equations (4.6) and (4.9) that, whereas the bit error probability of the sequential detector can be made arbitrarily close to that of the Viterbi detector by choosing suitably high thresholds $A_k \; \forall \; k \quad 0 \leq k \leq N_T$, we will by so doing, increase the value of the ASN undesirably. Also for the same value of additional probability of error introduced by the sequential detector to that of the fixed decoding delay decoder, there exist an infinite number of threshold settings and hence we seek in this section to find that optimum setting minimizing the ASN. For this we need to solve the equation:-

$$\frac{\partial}{\partial A_j}\left[\lambda PE_S + ASN\right] = 0 \qquad \forall \; j = 1,2,\ldots, N_T \qquad \ldots(4.10)$$

where PE_S corresponds to the additional probability of error term introduced in the bound on bit error probability in equation (4.6) by the sequential detector i.e.,

$$PE_S \leq \sum_{i=1}^{N_T} \sum_{j=N_F+1}^{N_o} G_i(j) \Bigg|_{D^d = \text{erfc}\left[\frac{\sqrt{d}}{\sigma_n} + \frac{\sigma_n \ln A_i}{2\sqrt{d}}\right]} \quad \ldots (4.11)$$

and to find the solution corresponding to a minimum of the ASN. This we do in Appendix 1 and the optimum thresholds turn out to be given by :-

$$A_j = \lambda \qquad \forall \; i = 1, 2, \ldots, N_T \; .$$

which is a very convenient result to have obtained from the practical point of view. The value of λ can of course be determined from the constraint on PE_S.

4.4.3 Buffer Requirement

The buffer requirement which stems from the fact that decisions are being made with variable decoding delay need only be as large as is necessary to make the probability of buffer overflow small compared to the bit error probability.

Now a large number of successive terminations at the later node levels will clearly cause a buffer overflow. Corresponding to each node level j, $i < j \leq i+N_T+1$ we can define a buffer requirement corresponding to the

possibility of successive terminations at this node level as follows. Let p(j) represent the probability of test termination at the $(i+j)^{th}$ node level. Then assuming independent termination of the sequential detection test for successive message symbols, we obtain that the probability of u successive terminations at the $(i+j)^{th}$ node level is given by $[p(j)]^u$. We are interested in u such that

$$[p(j)]^u \leq PE \qquad \ldots(4.12)$$

then the corresponding buffer requirement is given by

$\left[N_T + 1 + \frac{j}{R}(u-1) - u\right] n.\ q$ bits for the input buffer

and $u\left(\frac{N_T}{R} - 1\right)$ bits for the output buffer where:

n is the number of elements in the vector $\underline{r}_i$

q is the number of bits into which each element of $\underline{r}_i$ is quantized

and R is the ratio n_D/n_c of decoder clock rate to the input data rate. Clearly we need $R \geqslant ASN$.

Having obtained thus the corresponding buffer requirements for all j, we now consider the largest of these to be the required buffer memory since this would correspond to catering to the worst possible event.

Because at node level N_T+1, the test is forcibly terminated, it is expected that successive terminations at this node level will usually cause the greatest buffer requirement.

4.5. Results and Conclusions.

The bounds on the bit error probability and ASN of the sequential detector as well as the expression for the buffer memory derived in Section 4.4.3 have been evaluated for the case of rate 1/2 convolutional codes having constraint length K varying from 3 to 7 using the computer programme detailed in Appendix 2. The codes chosen were those which had the maximum value of the minimum free distance amongst codes of the same constraint length K. The generator polynomials of these codes (see [LAR - 73]) are given below :-

TABLE 5.1

K	Generator Polynomials
K = 3	$G_1(D) = 1 \oplus D \oplus D^2$ $G_2(D) = 1 \oplus D^2$
K = 4	$G_1(D) = 1 \oplus D \oplus D^3$ $G_2(D) = 1 \oplus D \oplus D^2 \oplus D^3$
K = 5	$G_1(D) = 1 \oplus D^3 \oplus D^4$ $G_2(D) = 1 \oplus D \oplus D^2 \oplus D^4$

K = 6	$G_1(D) = 1 \oplus D \oplus D^3 \oplus D^5$ $G_2(D) = 1 \oplus D^2 \oplus D^3 \oplus D^4 \oplus D^5$
K = 7	$G_1(D) = 1 \oplus D^2 \oplus D^3 \oplus D^5 \oplus D^6$ $G_2(D) = 1 \oplus D \oplus D^2 \oplus D^3 \oplus D^6$

In addition to the evaluation of these bounds, simulation results have also been obtained (using the computer program in Appendix 3) and the results presented in Fig. (5.1) to (5.6).

In Fig. (5.1) we show the variation of the PE with threshold setting for two values of the SNR obtained through simulation and evaluation of the bound. We see for moderate values of SNR the bound on PE to be fairly tight.

In Fig. (5.2) we show the variation of the ASN with the SNR for the case when the thresholds are set so as to make $PE_S = PE_{Viterbi}$ i.e. for the case when the bit error probability of the sequential detector is twice that of the Viterbi algorithm detector. We note firstly that the ASN is virtually independent of the SNR as seen from the simulation results and secondly that the ASN bound becomes tighter with increasing SNR.

In Fig. (5.3), we plot the variation of the ASN with the value of the constraint length K of the convolutional code using both the theoretical bound as well as the result obtained through simulation. Whereas the ASN is seen to monotonically decrease with K through the simulation results, the bound shows increasing K to have the same effect as an increase in the SNR namely, the bound becomes more tight for large values of the SNR.

In Fig. (5.4), we plot the variation of the ASN with the SNR, for the case when decision feedback is employed in this receiver using theoretical results in which we assume past decisions to be correct as well as simulation results. We see from this that there is no significant loss in performance arising out of decision feedback. The reason for this is that though decision feedback does result in error propagation (see below), and hence in an increase in the error probability, this is counter balanced by a decrease in ASN arising out of decision feedback.

In Fig. (5.5), we present the average number of additional errors caused by incorrect decision feedback for various values of the SNR with the threshold set so as to make the bit error probability comparable to that of the Viterbi detector. As expected, the number of successive

errors diminished with increase in the SNR.

In Fig. (5.6), we present the variation of the ASN with the ratio (PE_S / $PE_{Viterbi}$) where $PE_{Viterbi}$ denotes the bit error probability of the Viterbi detector. We observe from this that by adjusting the threshold A, we can obtain a wide range of performances from the sequential detector.

In Fig. (5.7), we present the variation of the maximum number of successive terminations at the $(N_T + 1)^{th}$ node level for different SNR's and we see that the result is not different from the approximation in Section 4.4.3 and that the number of successive terminations does not vary widely with the SNR.

In Table 5.2, we compare the storage requireme-nts of the sequential detector with that of the sequential detector. In this we include the input and output buffer requirements of the sequential detector, though we do not consider for either the survivor metric storage since it is common to both. The storage requirement of the sequen-tial of the sequential detector is given by :-

$$2^{(K-1)} + \left[N_T+1 + \frac{N_T}{R}(u-1) - u\right].n.q + u\left(\frac{N_T}{R} -1\right) \text{ bits}$$

where the symbols have the meaning given to them in Sec.4. .3

and the 1st term represents the survivor symbol storage of the sequential detector. We take $N_T = 4K$, $n = 2$ (corresponding to a rate 1/2 code), $q = 3$ (corresponding to 3 level quantization), $u = 3$ (as obtained from simulation results) and $R = ASN$ (corresponding to the smallest value of R which represents the worst possible case). We also present .. estimated values of the storage requirement of the sequential detector for the case $K = 8, 12$. From the table we see that for large values of K a tremendous saving in the storage requirement will result.

From these results we conclude that whereas the sequential detector does result in a considerable saving in the storage requirement, whether or not it can be applied in a particular situation will depend upon the nature of the situation i.e. upon whether a loss in decoding speed can be tolerated or not.

Table 5.2

STORAGE REQUIREMENT

K	Sequential Detector	Viterbi Detector
3	172 bits	48 bits
5	220 "	320 "
8	812 "	4096 "
12	2342 "	96 K bits

CHAPTER V

CONCLUSION

5.1 A Summary of the Thesis

In this thesis, binary sequential detection has been proposed for a class of information transmission systems. Associated with every sequential detector is a sequential detection test (SDT) which determines when a satisfactory decision regarding a particular message symbol can be made. Sequential detectors employing two different SDTs involving the a posteriori probabilities of the hypotheses H_i^o and H_i^1 and a maximum likelihood criterion respectively are considered in this report.

The sequential detector employing the a posteriori probabilities of the hypotheses H_i^o and H_i^1 is shown to have an error performance comparable to that of the optimum fixed decoding delay receiver. In addition to this, the sequential detector is shown to have a considerably reduced computational complexity though it does have a lower decoding speed. Though this sequential detector is believed to have a superior error performance when compared to the sequential detector employing the ML criterion, the former

is difficult to implement as it needs exponentiators in it implementation and it is for this reason that we consider the latter.

The sequential detector employing the ML criterion is shown to have an error performance comparable to that of the Viterbi decoder without requiring the large storage requirement of the latter. However, this sequential detector also suffers from a loss in decoding speed and hence its applicability in any situation will depend upon the nature of the particular situation. Upper bounds on the bit error performance as well as the average decoding delay have been obtained using an extension of Viterbi's generating function technique. These bounds have been evaluated for the particular case of a few convolutional codes. Finally, simulation results giving information as to the tightness of these bounds as well as shedding light on some important features of sequential detection are presented for the very same convolutional codes.

5.2 An Assessment of the two Sequential Detection Algorithms and Suggestions for Future Work.

Before an assessment of the sequential detectors presented in ChaptersIII and IV is made, a couple of remarks are made in connection with the applicability and advantages of applying sequential detection to communication systems.

Firstly, it should be noted that what really makes sequential detection applicable to the ITS model considered in Section 2.3 is the presence of the FSM in the model since this causes the received information r garding any particular message symbol to be spread over several symbols. Secondly, we would like to clarify the role played by sequential detection in reducing the hardware complexity of the fixed decoding delay receiver and the Viterbi detector. The reduction in hardware complexity is entirely due to the substitution of a serial implementation of the fixed decoding delay algorithms in place of the existing parallel implementation and has nothing whatever to do with sequential detection. Sequential detection serves the single purpose of improving the decoding speed of the receiver employing the serial implementation. From this it can probably be generalized that, if simplification of a receiver structure is described in the future, the first step would consist of discovering a serial implementation of the algorithm. It is only then that sequential detection can possibly be applied to improve, upon the decoding speed of the modified receiver structure.

With this, we will now go on to make an assessment of the sequential detectors employing the SPRT and the maximum likelihood (ML) criterion. The first point to be

noted in connection with the sequential detector employing the SPRT is the need for exponentiators in its implementation. If one goes beyond the exponentiators and considers the relative performance of the sequential detector and the optimum fixed decoding delay receiver, one finds that a considerable saving in the number of computational elements needed can be achieved by the application of sequential detection. The amount of saving can be computed from a knowledge of the parameters L and D of the sequential detector (see Section 3.6). However, for a meaningful assessment of the performance of the sequential detector, it is necessary to specify the ASN and the bit error probability of the detector in conjunction with each other since it is possible to trade off between the two by suitably adjusting the value of the threshold A. For this we need to know through some means, the value or at least an approximation to the value of the ASN for different values of the threshold A. Hence there is need for a more detailed analysis of this detector both in the way of determining the dependence of the ASN upon the threshold A as well as in determining the performance and saving in computational elements of the detector for specific situations of practical interest and future work in this area could be directed towards this topic.

We now move on to an assessment of the sequential detector employing the ML criterion. Unlike in the case of the sequential detector considered above, this detector can be implemented without much difficulty in practice. However, for any meaningful assessment of the performance of the sequential detector one must consider the average sample number in conjunction with the bit error probability of the detector just as in the case of the sequential detector employing the SPRT. Such an assessment can be made for any specific ITS as bounds on the ASN and bit error probability are available.

These bounds have been evaluated for the specific case of several convolutional codes and both theoretical as well as simulation bounds are presented in Fig. (5.1) to (5.6). From these results the following conclusions can be drawn :-

1. The theoretical bound obtained for the bit error probability is tight for moderate to large values of the SNR. The same is not true in the case of the ASN and the bound is conservative even for high values of the SNR.
2. The value of the ASN for the case when the bit error probability is comparable to that of the Viterbi detector is roughly given by half the constraint length. This can only be a rough guideline since strictly speaking, it

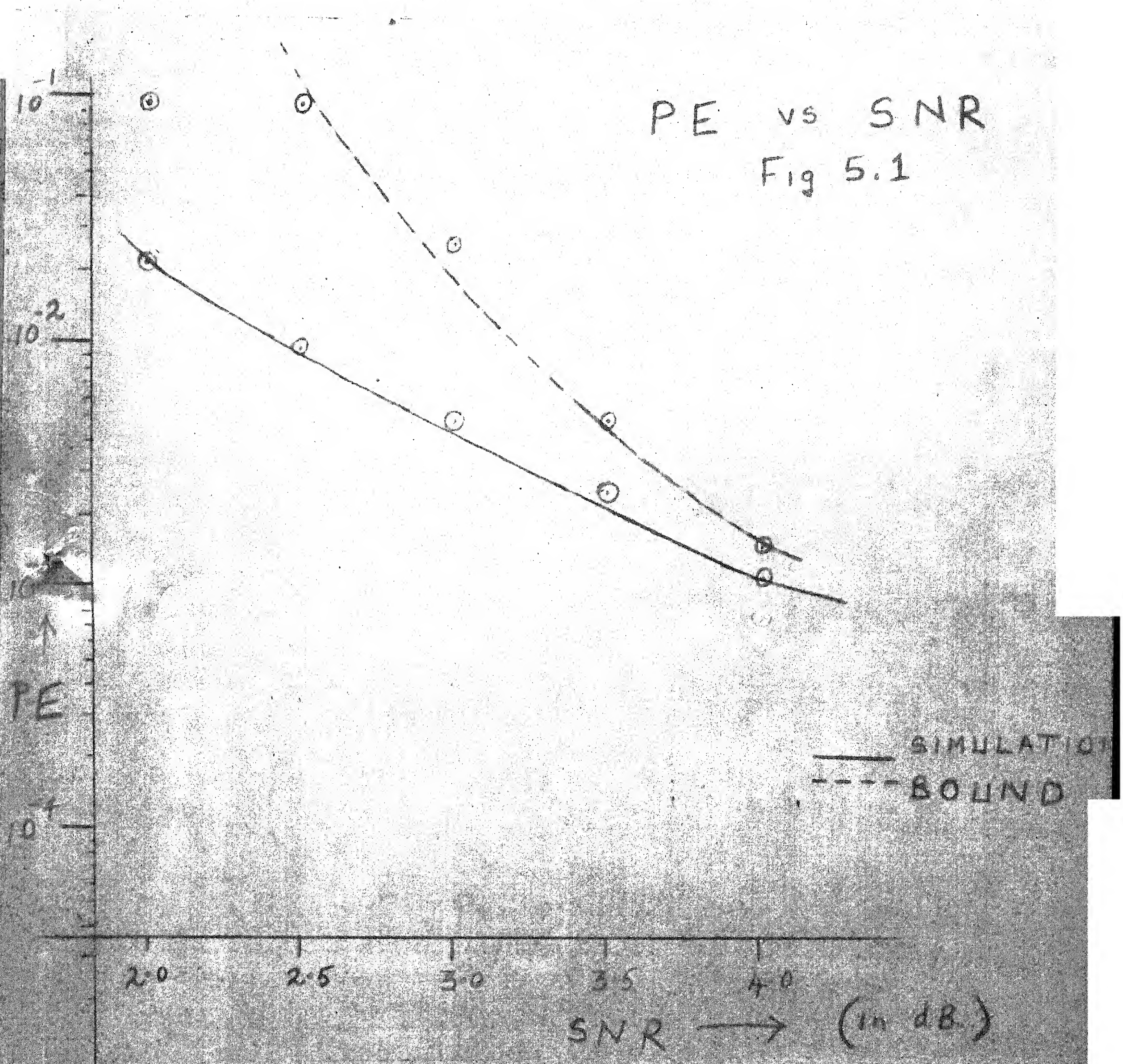
is the minimum free distance of the code and the number of code sequences having the minimum free distance as well as distances slightly greater than the minimum free distance which effect the value of the ASN rather than the value of the constraint length K.

3. Decision feedback does not result in any serious loss in performance and can therefore be gainfully employed to dispense with the need for the dormant storage section (see Fig. (4.1)).

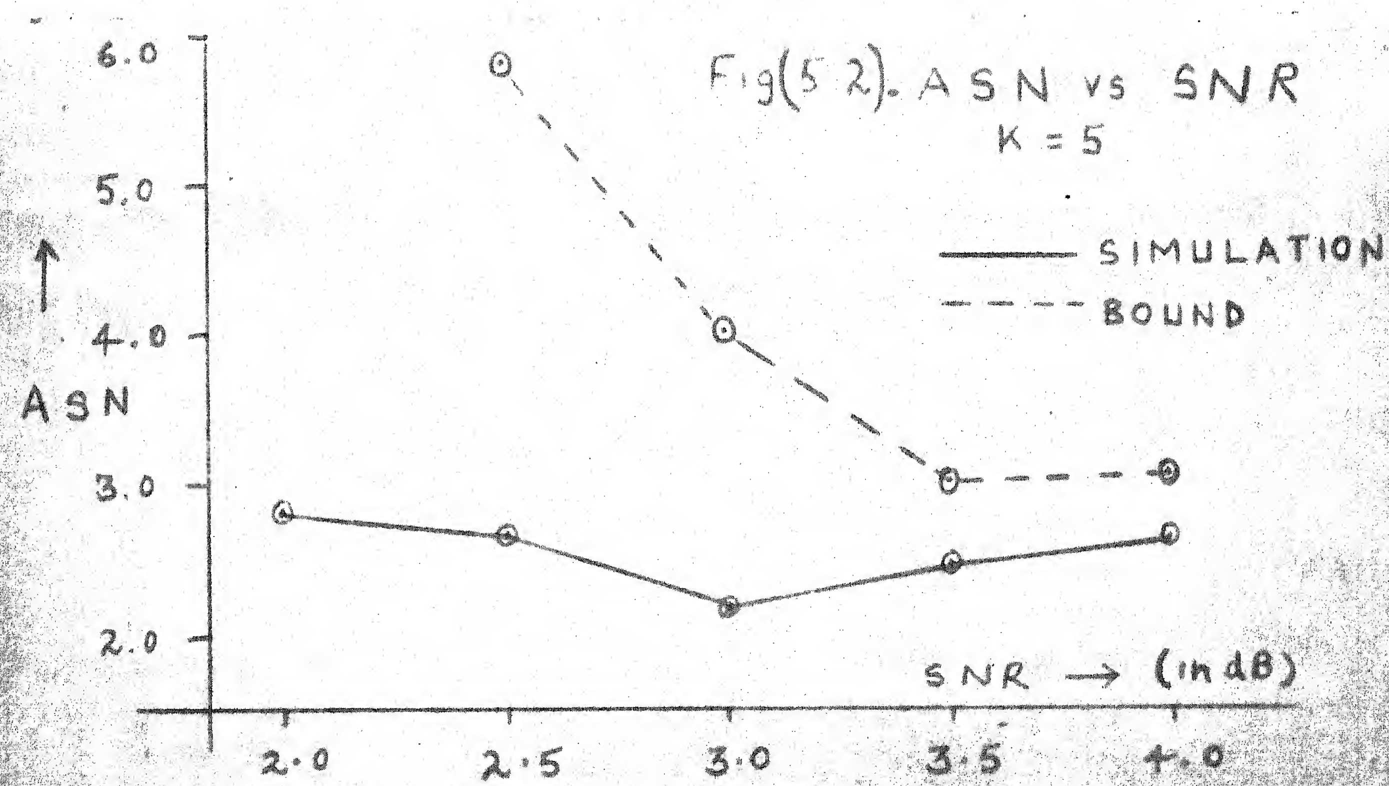
4. The buffer requirement of the sequential detector is not very much greater than that determined in Section 4.4.3 under the assumption of independant termination of the SDT at any node level of the trellis for successive message symbols.

All these observations hold good only for the case of the example of convolutional codes and it is not possible to generalize just on the basis of these results. Hence, evaluation of these bounds for the case of i.s.i. or any other practical example of such an ITS would be of interest and future work could be directed towards this area.

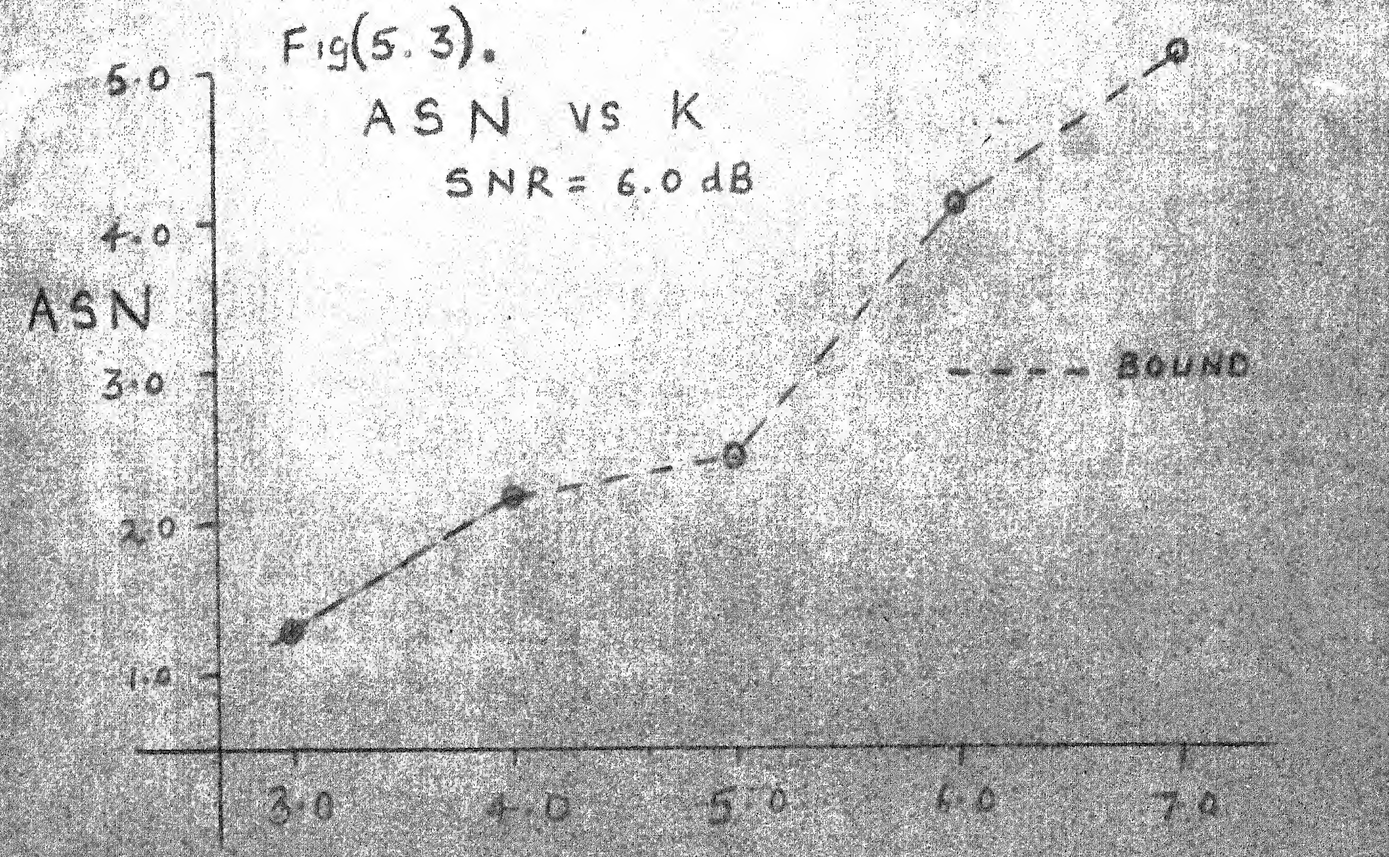
Hence we see that whereas the saving in computational complexity offered by both sequential detectors



PE vs SNR
Fig 5.1
PE
SNR → (in dB.)
2.0
2.5
3.0
3.5
4.0
SIMULATION
BOUND



Fig(5.2). ASN vs SNR
K = 5
SIMULATION
BOUND
ASN
6.0
5.0
4.0
3.0
2.0
SNR → (in dB)
2.0
2.5
3.0
3.5
4.0



Fig(5.3).
ASN vs K
SNR = 6.0 dB
ASN
BOUND
5.0
4.0
3.0
2.0
1.0
3.0
4.0
5.0
6.0
7.0

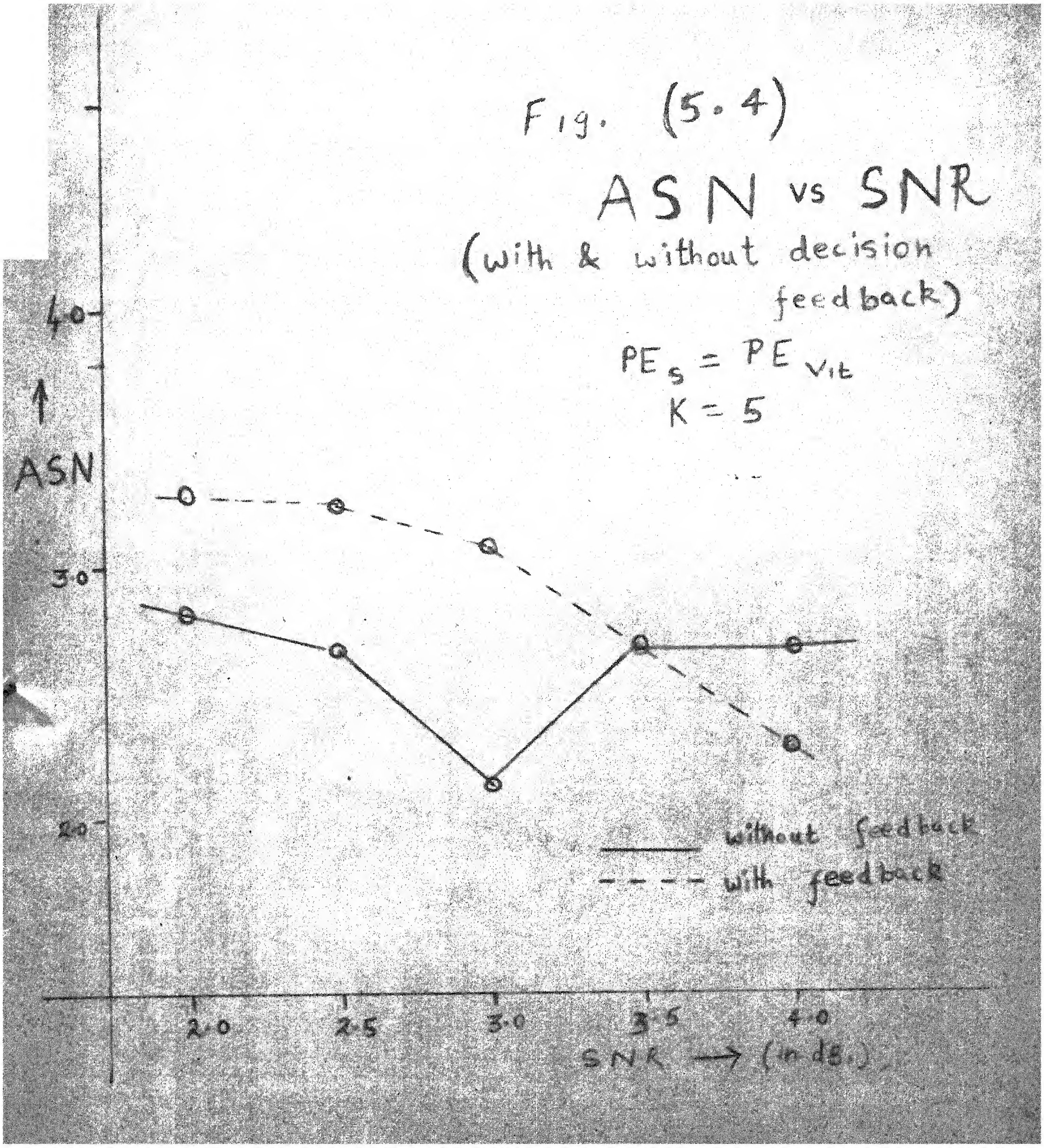


Fig. (5.4)
ASN vs SNR
(with & without decision feedback)
$PE_s = PE_{Vit}$
K = 5
ASN
4.0
3.0
2.0
2.0
2.5
3.0
3.5
4.0
SNR → (in dB.)
without feedback
with feedback

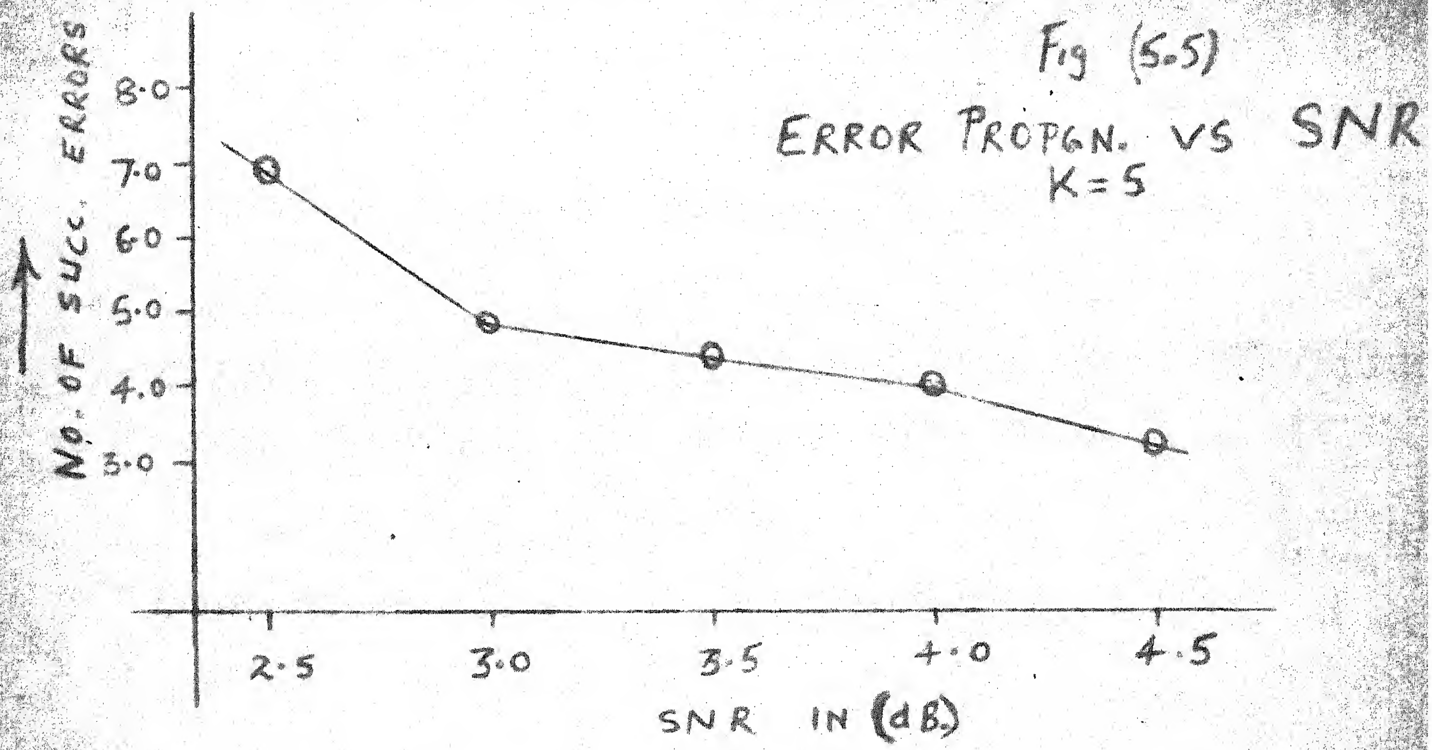


Fig (5.5)
ERROR PROPGN. VS SNR
K=5

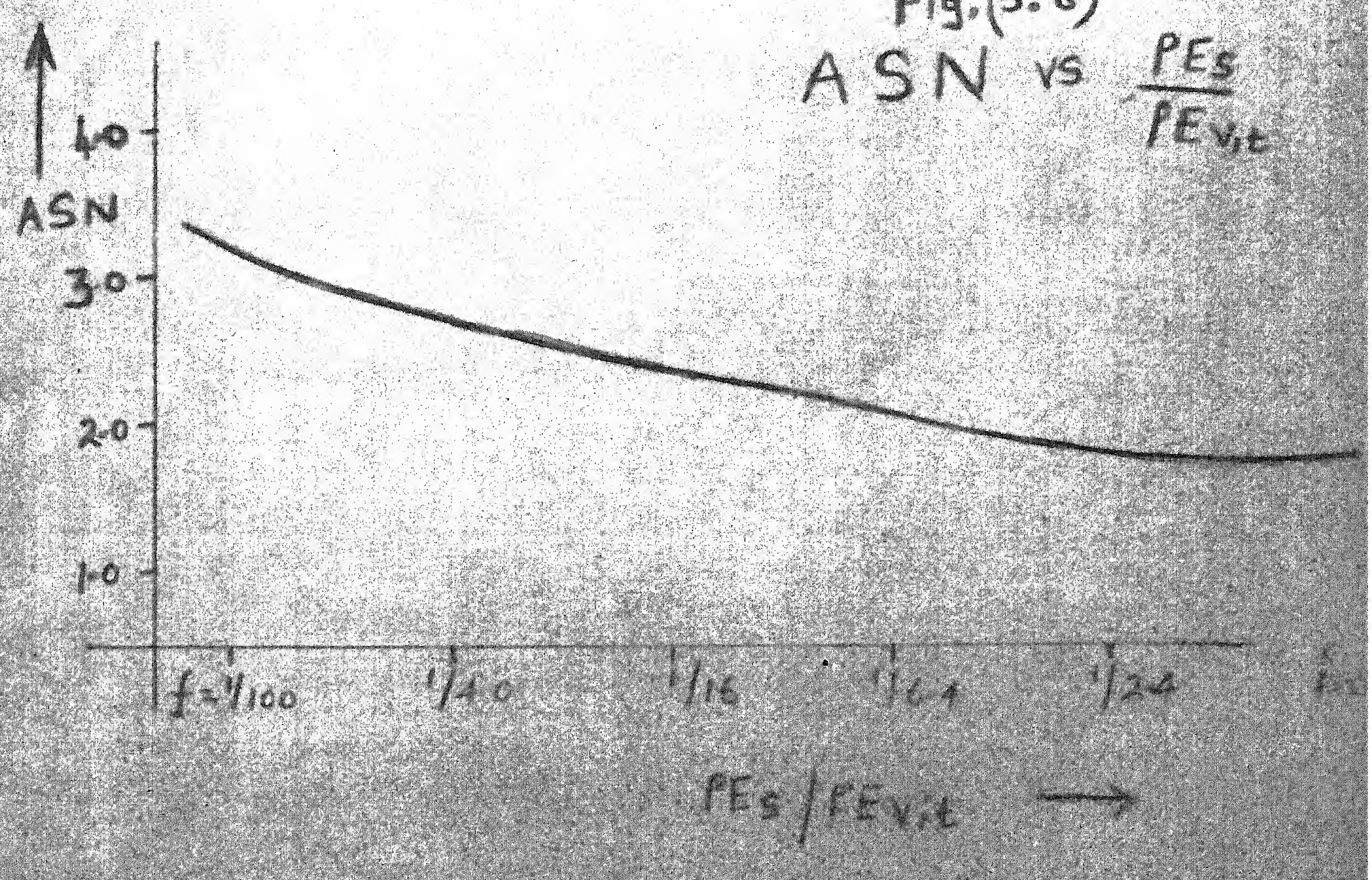


Fig. (5.6)
ASN VS PEs / PEvit

is indisputable, the performance of the sequential detector in terms of the ASN and bit error probability have to be determined afresh for any particular ITS for one to be able to make a decision as to whether or not to employ sequential detection.

Finally, an interesting area towards which future work could be directed is the consideration of other sequential detection tests for the sequential detector.

APPENDIX I

From equation (4.10) we see that in order to determine the optimum threshold setting we need to solve the set of N_T equations :

$$\frac{\partial F}{\partial A_j} = 0 \quad \forall\; j=1,2,\ldots,N_T \qquad \ldots\ldots \text{(A1.1)}$$

where $F = ASN + \lambda PE_S$, λ being the Lagrange multiplier.

However we do not have with us an expression for the function F in terms of the threshold A_j and hence we substitute for the ASN and PE_S, the bounds obtained in equations (4.6) and (4.9) which will amount to the same thing if the bounds are tight.

For the sake of convenience, we reproduce here below the bounds referred to above :

$$PE_S \leq \sum_{j=1}^{N_T} \sum_{k=N_F+1}^{N_c} G_{i+j}(k) \Big|_{D^d = \operatorname{erfc}\left[\frac{\sqrt{d}}{\sigma_n} + \sigma_n \frac{\ln A_j}{2\sqrt{d}}\right]}$$

$$\ldots\ldots \text{(A.1.2)}$$

$$ASN \leq 1 + \sum_{j=1}^{N_T} \sum_{k=N_F+1}^{N_c} G_{i+j}(k) \Big|_{D^d = \operatorname{erfc}\left[\frac{\sqrt{d}}{\sigma_n} - \frac{\sigma_n \ln A_j}{2\sqrt{d}}\right]} + \sum_{j=1}^{N_T} \sum_{k=N_F+1}^{N_c} G_{i+j}(k) \Big|_{D^d = \operatorname{erfc}\left[\frac{\sqrt{d}}{\sigma_n}\right]} \qquad \ldots\ldots \text{(A.1.3)}$$

Hence we can rewrite equation (A.1.1) using equations (A.1.2) and (A.1.3) as follows :

$$\frac{\partial}{\partial A_j}\left\{ \lambda \cdot \sum_{k=N_F+1}^{N_c} G_{i+j}(k) \Big|_{D^d = \operatorname{erfc}\left[\frac{\sqrt{d}}{\sigma_n} + \frac{\sigma_n \ln A_j}{2\sqrt{d}}\right]} + \sum_{k=N_F+1}^{N_c} G_{i+j}(k) \Big|_{D^d = \operatorname{erfc}\left[\frac{\sqrt{d}}{\sigma_n} - \frac{\sigma_n \ln A_j}{2\sqrt{d}}\right]} \right\} = 0 \qquad \text{..... (A.1.4)}$$

Now, $\sum_{k=N_F+1}^{N_c} G_{i+j}(k)$ is clearly a polynomial in D and therefore we can write :

$$\sum_{k=N_F+1}^{N_c} G_{i+j}(k) = \sum_d C_d . D^d$$ where $C_1, C_2, \ldots,$ are coefficients of the polynomial in D.

Equation (A.1.4) can be written now as :

$$\frac{\partial}{\partial A_j}\left\{ \sum_d C_d \left(\lambda . \operatorname{erfc}\left[\frac{\sqrt{d}}{\sigma_n} + \frac{\sigma_n}{2} \frac{\ln A_j}{\sqrt{d}}\right]\right) \right\} + \frac{\partial}{\partial A_j}\left\{ \sum_d C_d \left(\operatorname{erfc}\left[\frac{\sqrt{d}}{\sigma_n} - \frac{\sigma_n \ln A_j}{2\sqrt{d}}\right]\right)\right\} = 0$$

i.e. $$\sum_d C_d \left\{ \frac{\lambda \cdot (-1)}{\sqrt{2\pi}} \cdot e^{-\frac{1}{2}\left[\frac{\sqrt{d}}{\sigma_n} + \frac{\sigma_n \ln A_j}{2\sqrt{d}}\right]^2} \cdot \left(\frac{\sigma_n}{2\sqrt{d}}\right) \cdot \frac{1}{A_j} \right.$$

$$\left. + \; (-1) \cdot \frac{1}{\sqrt{2\pi}} \cdot e^{-\frac{1}{2}\left[\frac{\sqrt{d}}{\sigma_n} - \frac{\sigma_n \ln A_j}{2\sqrt{d}}\right]^2} \left(-\frac{\sigma_n}{2\sqrt{d}}\right) \cdot \frac{1}{A_j} \right\} = 0$$

i.e., $$\sum_d C_d \cdot \left(\frac{\sigma_n}{2\sqrt{d}}\right) \cdot \frac{1}{A_j} \cdot \frac{1}{\sqrt{2\pi}} \cdot e^{-\frac{1}{2} \cdot \left(\frac{d}{\sigma_n^2}\right)} \cdot$$

$$\cdot \, e^{-\frac{1}{2}\left[\frac{\sigma_n}{2} \frac{\ln A_j}{\sqrt{d}}\right]^2} \cdot \left\{ e^{\frac{1}{2} \ln A_j} - \lambda e^{-\frac{1}{2} \ln A_j} \right\} = 0$$

and hence we see that for $A_j = \lambda$, $\frac{\partial F}{\partial A_j} = 0 \;\; \forall \;\; j = 1,2,\ldots,N_T$.

Examining the 2nd derivative at this point reveals that :

$$\frac{\partial^2 F}{\partial A_j^2} = \sum_d C_d \left[\underbrace{\left(\frac{\sigma_n}{2\sqrt{d}}\right) \cdot \frac{1}{\sqrt{2\pi}} \cdot e^{-\frac{1}{2}\frac{d}{\sigma_n^2}}}_{>0} \right] \cdot$$

$$\left[\left\{ e^{\frac{1}{2} \ln A_j} - \lambda e^{-\frac{1}{2} \ln A_j} \right\} \cdot \left\{ \frac{\partial}{\partial A_j} \left(\frac{1}{A_j} \cdot e^{-\frac{1}{2}\left(\frac{\sigma_n \ln A_j}{2\sqrt{d}}\right)^2} \right) \right\} \right.$$

$$\left. + \left\{ \frac{1}{A_j} \cdot e^{-\frac{1}{2}\left(\frac{\sigma_n \ln A_j}{2\sqrt{d}}\right)^2} \right\} \cdot \left\{ \frac{\partial}{\partial A_j} \left(A_j^{1/2} - \frac{\lambda}{A_j^{1/2}} \right) \right\} \right]$$

$$= \sum_{d=1,2,\ldots} C_d \left(\frac{\sigma_n}{2\sqrt{d}}\right) . \left(\frac{1}{\sqrt{2\pi}}\right) . e^{-\frac{1}{2}\frac{d}{\sigma_n^2}} .$$

$$. \left[\frac{1}{2} . e^{-\frac{1}{2}\left(\frac{\sigma_n \ln A_j}{2\sqrt{d}}\right)^2}\right] . \left[\frac{1}{2A_j^{1/2}} + \frac{\lambda}{2A_j^{3/2}}\right] > 0.$$

unless $C_d = 0$ for all d.

Hence the solution $A_j = \lambda$, $\forall$ $j = 1,2,\ldots, N_T$ corresponds to a minimum of the function F.

APPENDIX 2

PROGRAMME TO EVALUATE THE BOUNDS ON ASN AND PE OF THE SEQUENTIAL DETECTOR EMPLOYING THE ML CRITERION.

In this section of the report, we consider a programme written to evaluate the bounds obtained in Chapter IV on the ASN and the average bit error probability of the sequential detector employing the ML criterion. when used for the detection of convolutionally encoded data received over AWGN channels. The programme has been written to evaluate the bounds for rate 1/2 convolutional codes of any constraint length. Evaluation of these bounds is carried out by using an extension of the generating function technique introduced by Viterbi [VIT -71]. The proceedure that will be followed has already been outlined in Section 4.3 and we only point out here a few simplifications as well as a modification in this proceedure. The modification arises because we are in a position to tighten the bit error probability bound with the foreknowledge that the optimal threshold setting envisages uniform thresholds $A_j \;\forall\; j \;,\; 1 \leq j \leq N_T$. The simplifications arise only because we are dealing with the particular case of convolutional codes.

The first simplification arises since convolutional codes are group codes and hence the bit error probability is the same for every transmitted sequence. Clearly it is no longer necessary to average the bit error probability over the ensemble of all possible transmitted sequences. Hence in determining the bound on the bit error probability, we deal with the state and trellis diagrams instead of the error state and error trellis diagrams respectively. We consider arbitrarily the all zero sequence to be the true message sequence throughout the remainder of this section.

We will now consider how the bound on the probability of error(bit error) may be tightened given that the thresholds $A_j \; \forall \; j \;, 1 \leq j \leq N_T$ are equal. Consider a branching message sequence which branches off at a node level at or prior to the i^{th} node level and is such that at some node level $i+j$ where $1 \leq j \leq N_T$, it has a metric exceeding that of the true message sequence by an amount equal to $\ln A_j$. Also, let the sequence be such that the output of the FSM corresponding to its $(i+j-1)^{th}$ symbol is a stream of zeroes. Then clearly, the sequence obtained by truncating the last message symbol of the sequence will also have a metric exceeding that of the 1st $(i+j-1)$ symbols of the true message sequence by an amount equal to $\ln A_j$.

Hence when we sum the probability of these two events (which entirely coincide) in order to obtain the union bound, we are indulging in some double counting which we would like to avoid. This we can do very simply by ignoring one or the other of the two sequences in such cases and in the program the longer sequence is ignored.

The next modification we consider here has the advantage of decreasing computation time on the computer. In order to obtain the weight function vector $G_{i+1}]$ from the weight function vector $G_i]$, we will need to perform the matrix multiplications indicated in equation 4.1 which we reproduce below :

$$G_{i+1}] = [W] . G_i]$$

The number of components of the vector $G_i] = N_F$ for the case of the convolutional encoder since we are dealing with the trellis and not the error trellis diagram for this FSM. Hence in the ordinary way, this matrix multiplication would involve N_F^2 multiplications. However, a close inspection of the weight function matrix $[W]$ for the case of rate 1/2 convolutional codes reveals that since any node at a given node level can be reached from only two nodes at the previous node level, there are only two non-zero terms in each row of the matrix. To illustrate this, the

trellis and weight function matrix for the rate 1/2, K = 3 convolutional code considered earlier in Chapter IV is shown in Figs. (A.2.1a) and (A.2.1b). Hence, a reduction in the number of multiplications from N_F^2 to $2.N_F$ can be achieved if we could determine for every node level j at the $(i+1)^{th}$ node level, the two nodes leading to it from the i^{th} node level. To do this, we will need to number the nodes in some fashion. The state of the encoder is determined by the past (K-1) inputs received by it. Let the state be represented therefore by a binary number with the latest input forming the least significant bit. Now, consider the decimal number obtained by adding 1 to the above binary number and let the nodes of the trellis diagram which are nothing but states of the FSM encoder, be numbered by their corresponding decimal numbers. Then we see that in the trellis diagram the nodes j and $j+\frac{N_F}{2}$ lead to the nodes (2j-1), 2j for all j, $1 \leq j \leq N_F/2$. This suggests that we divide the nodes at two adjacent node levels into sets of 4 nodes as in Fig. (4.C). In the jth such set of 4 nodes, we will refer to the nodes whose numbers are j, $j+\frac{N_F}{2}$, 2j -1, 2j as the nodes NA(j), NB(j), NC(j) and ND(j) respectively. As shown in Fig. (A.2.1c), associated with every such set of 4 nodes, we have 4 weight functions L AC(j), L AD(j),

LBC(j) and LBD(j). It is clear that in order to carry out the matrix multiplication of eqn. 2.11 using 2 . N_F multipliers, we only need to know the weight functions LAC(j), LBC (j) , LAD(j) and LBD(j) for each value of j. In the program, these values for all j , $1 \leq j \leq N_F/2$ are initially evaluated and stored so that they can be recalled for each matrix multiplication.

The last simplification that is possible arises as follows. As we have already mentioned before, the probability of error in detecting the symbol x_i being caused by a message sequence having i^{th} symbol = 1 and whose corresponding transmitted sequence is at a Eucledean distance 2d from the actually transmitted sequence is given by :-

$$\mathrm{erfc}\left[\frac{\sqrt{d}}{\sigma_n} + \frac{\sigma_n \ln A}{2\sqrt{d}}\right]$$

where σ_n^2 is the variance of the noise at the output of the channel.

The Eucledean distance 2 d between the transmitted sequence corresponding to an output sequence having a Hamming distance 'p' from the all zero output sequence and the transmitted sequence corresponding to the all zero sequence itself is given by :-

$$2d = \sqrt{4\,ph^2}$$

$$\text{i.e.} \quad 4d^2 = p.\,(4h^2) \qquad \text{...(A.2.1)}$$

Hence the square of the Euclidean distances corresponding to the various message sequences can take on only discrete values and it is sufficient to know the value of the Hamming distance p corresponding to every such sequence in order to evaluate the probability that its metric will exceed that of the true message sequence by any specified amount. Hence we use D^p as the weight function instead of the function D^{d^2}.

From this, it is evident that the weight functions of the various nodes at any node level are polynomials in D.

In the program therefore, we store the weight functions in shift registers, maintaining one shift register for each node in which the p^{th} bit gives the number of message sequences leading to that node which are such that their output sequences are at a distance 'p' from the all zero sequence . The procedure for evaluating a bound on the probability of error in detecting the message symbol x_1 involves recursive evaluation of the weight functions at successive node levels. This recursive procedure now clearly involves multiplication of polynomials in D. Since

the coefficients of the polynomials are stored in shift registers, clearly multiplications correspond to "shifting" of coefficients within the shift register. Hence, when we wish to obtain the weight functions of the node NC(j) at a node level from those of the nodes NA(j) and NB(j) at the previous node level, we shift the coefficients of the polynomials giving the weight functions of the nodes NA(j) and NB(j) by amounts equal to LAC(j) and LBC(j) respectively and add the resultant polynomials.

It remains to explain the programme itself. The Flow chart shown in Fig. A.2. is easy to follow. The 1st step in the programme in determining the values of LAC(j), LBC(j), LAD(j), LBD(j), for various values of j, $1 \leq j \leq N_F/2$. Thereafter the recursive process involved in obtaining the weight functions starting from the $(i-N_S)^{th}$ to the $(i-N_T)^{th}$ node level is begun. In this programme N_S has been somewhat arbitrarily chosen equal to N_T. In the process three different weight functions are used as already explained in Section 4.3. The 1st in going from the $(i-N_S)^{th}$ node level to the i^{th}, the second in advancing to the $(i+1)^{th}$ node level from the i^{th} node level and the last in advancing from the $(i+1)^{th}$ node level to the $(i+N_T+1)^{th}$ node level. After each recursive computation, beyond node level i and upto

node level $(i + N_T)$ the sum of the weight functions of all the nodes excepting all zero node at each node level is added to the contents of a register RASN which at the end of the recursive computation will contain the coefficients of the polynomial used in obtaining the bound on the ASN. This corresponds to an evaluation of the 2nd term in equation (4.2). The weight functions of the all zero node at node levels ranging from the $(i+1)^{th}$ to the $(i+N_T)^{th}$ node level are added and stored in a register RVIT together with the weight functions of all the nodes at the $(i+N_T+1)^{th}$ node level. At the end of the recursive computation, this register will contain the coefficients of the polynomial in D needed to obtain the bound on the probability of the practical Viterbi detector. This corresponds to evaluation of the last 2 terms in the RHS of equation (4.7).

At the end of each step in the recursive computational beyond the i^{th} node level and upto the $i+N_T{}^{th}$ node level, the weight functions corresponding to sequences of the type discussed above in regard to a tightening of the bound on the bit error probability are subtracted from the sum of weight functions stored in register RASN and the polynomial thus obtained used to obtain the additional probability of bit error PE_S introduced by the sequential detector. This corresponds to improving upon the bound for PE_S given in Eqn. (4.11).

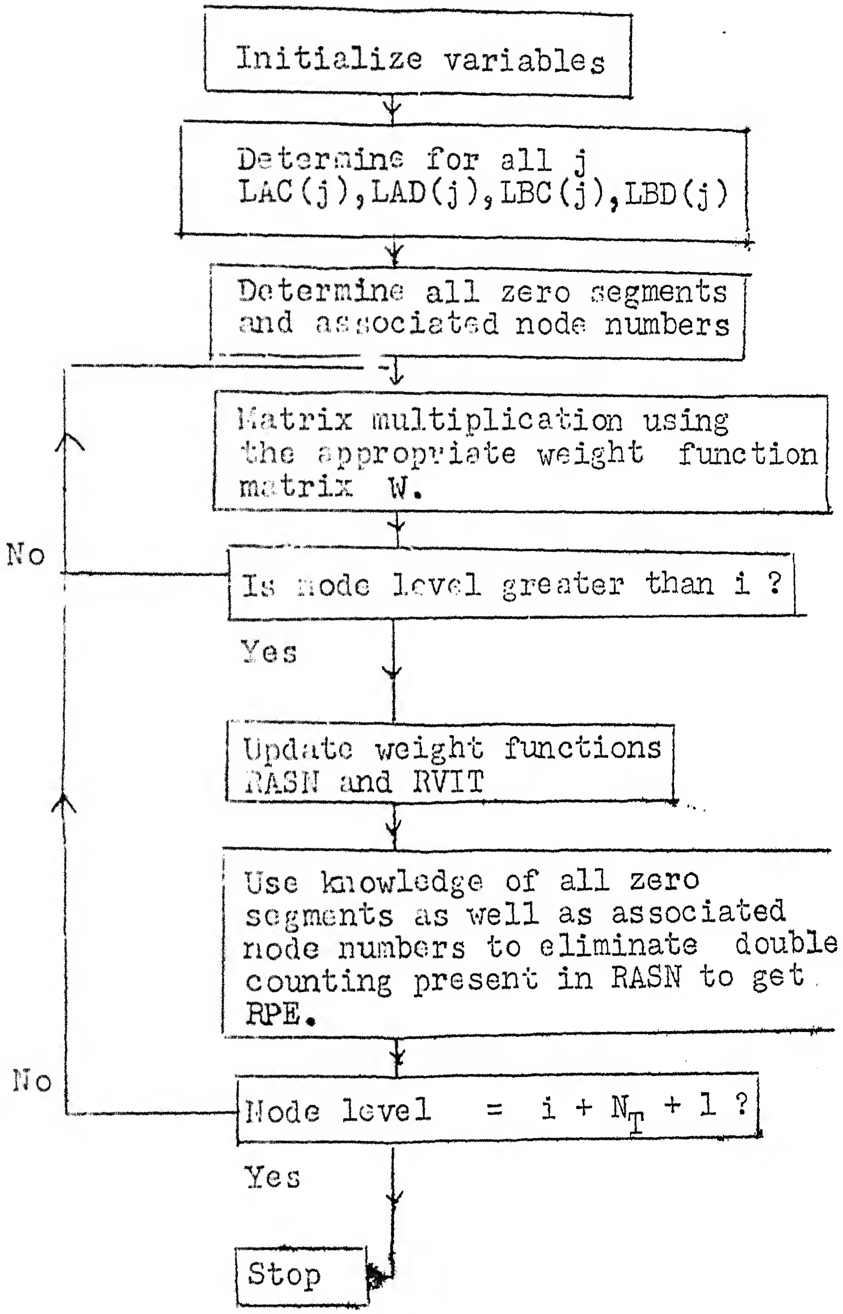


Fig. A.2.1. Flow chart for program evaluating bound on PE_S and ASN using generating function technique.

APPENDIX 3

In this section of the report, we present and explain a programme written to simulate the decoding of convolutionally encoded data when received over AWGN channels using the sequential detector employing the ML criterion for its SDT. The programme has been written for rate 1/2 convolutional codes having any constraint length. Since convolutional codes are group codes which have the property that the probability of error is the same for each of the transmitted sequences, we have chosen the all zero sequence to be the transmitted sequence in this programme. The AWGN introduced by the channel has been simulated by generating Gaussian random variates using the Box Muller technique [NAR-78] .

In the following, we present the steps followed in the programme for the case when the first future symbol $\underline{r}_1$ corresponding to a message symbol x_i is examined followed by those for the case when subsequent future symbols $\underline{r}_{i+1}$, $\underline{r}_{i+2}$, ..., $\underline{r}_{i+j}$, ... are being examined by the detector. It will be seen that some of the steps followed are common to both cases and hence the two proceedures are treated separately only for the sake of ease

in understanding. Steps C1 to C11 below are those followed in the case when the 1st future symbol $\underline{r}_i$ has just been examined.

C1. The matrix of the survivors at the i^{th} node level are transferred from the dormant storage where they were placed during the detection process involving the symbol x_{i-1} to the active storage section.

C2. A new received symbol $\underline{r}_{i+N_T}$ is generated.

C3. The metrics of the segments linking the i^{th} to the $(i+1)^{th}$ node level are computed using the received symbol $\underline{r}_i$.

C4. The metrics of the survivors at the $(i+1)^{th}$ node level are determined from those of the survivors at the ith node level using the above segment metrics and a set of add compare select operations.

C5. The i^{th} message symbol of each survivor at the $(i+1)^{th}$ node level is stored alongside its metric in the active storage section.

C6. The metrics of these survivors are also stored in the domant storage section as they will be needed for the detection of the symbol x_{i+1}.

C7. The survivor with greatest metric amongst survivors having i^{th} message symbol = 0 and = 1 are determined. Using these and the threshold A, the SDT is carried out.

C8. If the test results in a decision, the programme statistics are updated. These include the latest value of the ASN, the maximum number of consecutive terminations at the terminating node level N_T as well as the bit error probability of the sequential detector.

C9. If the test does not result in a decision the sequence of steps D1, ... are carried out.

The following steps D1 to D7 are followed in the case when the detection process of the message symbol x_i is continued and an additional received symbol $\underline{r}_{i+j}$ examined. $1 \leq j \leq N_T$.

D1. The segment metrics linking the survivors at the i+j and $(i+j+1)^{th}$ node level are determined using the received symbol $\underline{r}_{i+j}$.

D2. The metrics of the survivors at the $(i+j+1)^{th}$ node level are determined using the above segment metrics and a set of ACS operations.

D3. The i^{th} message symbol of each survivor at the $(i+j+1)^{th}$ node level is determined from the message symbol of the survivor at the $(i+j)^{th}$ node level leading to it.

D4. The metrics of the survivors S^o_{i+j+1} and S^1_{i+j+1} are then determined and the SDT carried out using these metrics and the threshold A. If $j = N_T$, a unity threshold is used.

D5. If the test results in a decision, the program statistics referred to in step C8 are updated and the next step followed is D7.

D6. If the test does not result in a decision, the above steps D1 to D6 are repeated with the next received symbol r_{i+j+1} and so on

D7. If a sufficient number of symbols x_i , $i = 1,2,\ldots$ have been detected to ensure that the statistics computed are reasonably accurate, the programme is terminated. Otherwise the detection of the next symbol x_{i+1} is begun at step C1.

Fig. A.3.1 shows the flow chart for the programme in which the common steps followed in the two cases are clearly brought out.

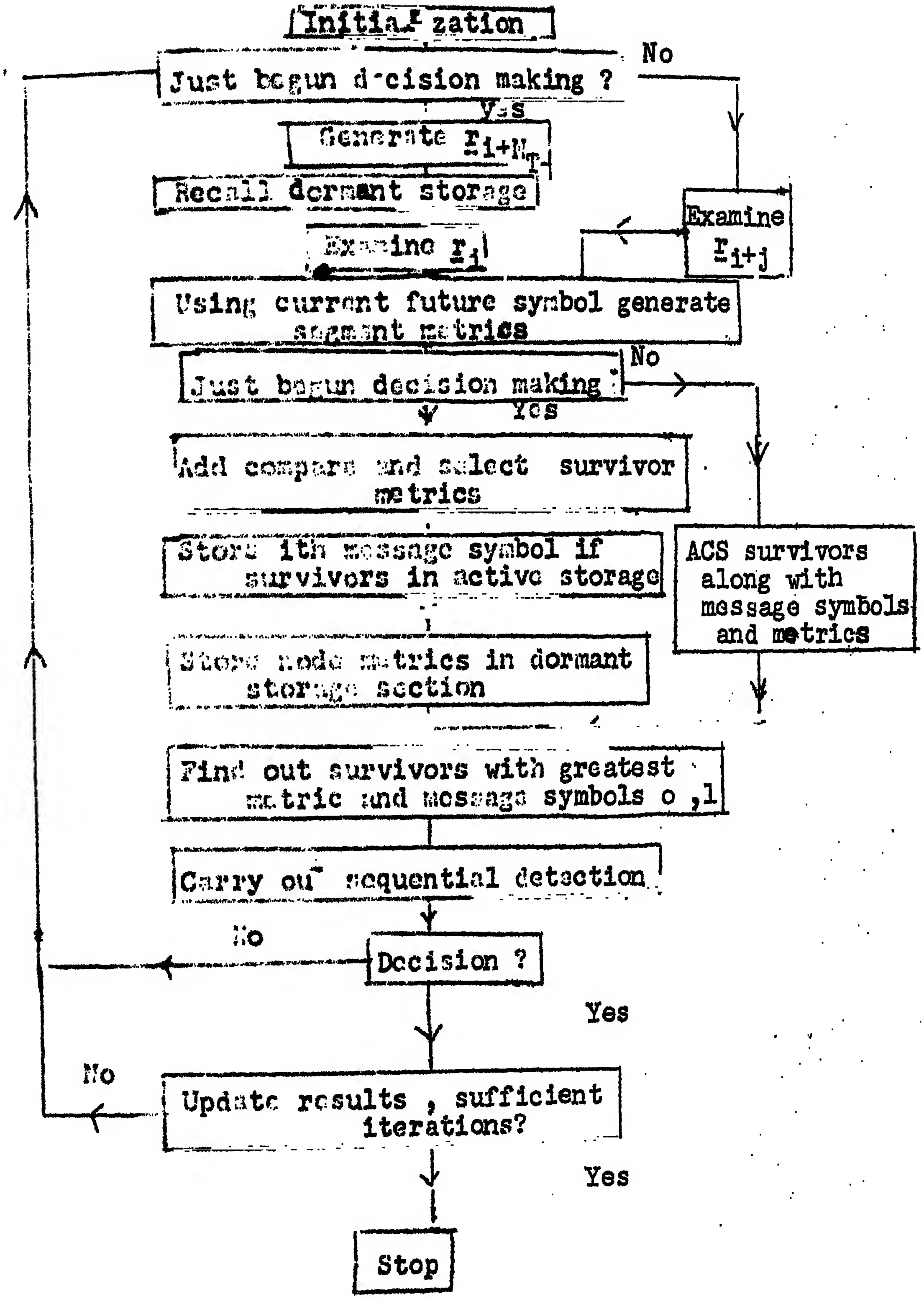


Fig. A.3.1 Flow chart for simulation of sequential detector using ML criterion for SDT.

```
C AS FOLLOWS:- DATA,DIMENSION,FORMAT FOR READING TAP GAINS.
C THIS PROGRAMME EVALUATES THE BOUND ON THE PE AND ASN OF A SEQ. DET
C USING AN EXTN. OF THE GEN. FN. TECHNIQUE INTRODUCED BY VITERBI.
C THE PROG. HAS BEEN WRITTEN TO TACKLE GENERAL CON. CODES HAVING RAT
C LIST OF VARIABLES:-
C K=KL=CONSTRAINT LENGTH OF THE CODE,RATE OF CODE=1/NRAT,
C NT IS THE TEMINATING NODE LEVEL=4K IN GENERAL,
C NODES=NO. OF NODES IN THE TRELLIS=2**(K-1)
C TAP(I,J) I=1,2 J=1,2,...K  GIVES THE GEN. POLYNOMIALS
C NITER=NO. OF RECURSIVE EVALUATIONS OF THE WT. FN. VECTOR
C III=RUNNING VAR. GIVING NO. OF RECURSIVE COMPUTATION
C LNTT GIVES THE MAX. POSSIBLE DISTANCE OF A SEQUENCE
C AT THE END OF PRESENT RECURSIVE COMPUTATION.
C A(I,J) I=1,2...,NODES , J=1,2,....LNTT      REFERS TO THE
C WEIGHT FNS. OF NODES AT BEGINNING OF PRESENT COMPUTATION
C B(I,J) I=1,2,...,NODES,  J=1,2,...,LNTT,    REFERS TO THE WEIGHT
C FNS. AT THE END OF THE COMPUTATION.
C RASN(I), I=1,2,....,LNTT GIVES THE SUM OF THE WEIGHT FNS.NEEDED TO
C TO EVALUATE ASN BOUND.
C RPE(I), I=1,2,...LNTT GIVES THE SUM OF WEIGHT FNS. USED TO EVALUAT
C THE BOUND ON THE ADDITIONAL PE OF THE SEQ.DET.
C RVIT(I), I=1,2,...LNTT GIVES THE SUM OF WEIGHT FNS. CONSTITUTING
C A BOUND ON THE PE OF A VIT. DET.
C LAC(I),LAD(I),LBC(I),LBD(I) I=1,2,...,2**(K-2)  GIVE THE
C DISTANCES OF SEGMENTS AC(I),AD(I),BC(I),BD(I) OF THE TRELLIS.
C    ********************** END **********************
        DIMENSION TAP(10,10),A(64,120),B(64,120),JJ(10),E(10),F(10),
       A G(10),H(10),NE(10),NF(10),NG(10),NH(10),NZER(64),D(100),C(10(
        DIMENSION LAC(32),LBC(32),LAD(32),LBD(32)
        DIMENSION RPE(120),RASN(120),RVIT(120)
          DOUBLE PRECISION  DUMS,DUM1
          OPEN(UNIT=21,FILE='V.DAT')
        READ(21,10)KL,NRAT,NT
   10 FORMAT(2I4,I5)
        TYPE11,KL,NRAT,NT
   11 FORMAT(5X,2I4,I5,'KL,NRAT,NT')
        KL1=KL-1
         KL2=KL-2
        NODES=2**(KL-1)
         NTT=NT*NRAT
        KNTT=KL*NRAT
        NTL1=NT-1
          NTD=NT*2
          NTP1=NT+1
          NTD1=NTD+1
          NITER=2*NT+1
C INITIALIZATION
          DO21=1,NTT
        RASN(I)=0.0
        RPE(I)=0.0
          RVIT(I)=0.0
    2 CONTINUE
        NDBY2=NODES/2
          NDB2P4=NDBY2+4
C TO READ IN THE TAPPINGS OF THE ENCODER
        READ(21,20)((TAP(I,J),J=1,KL),I=1,NRAT)
   20 FORMAT(7F4.1)
        IZ=1
```

```
      JJ(2)=I2-1
      IF(KL-4)40,50,41
   41 DO40I3=1,2
      JJ(3)=I3-1
      IF(KL-5)40,50,42
 42   DO40I4=1,2
      JJ(4)=I4-1
      IF(KL-6)40,50,43
   43 DO40I5=1,2
      JJ(5)=I5-1
      IF(KL-7)40,50,44
   44 DO40I6=1,2
      JJ(6)=I6-1
      IF(KL-8)40,50,45
   45 DO40I7=1,2
        JJ(7)=I7-1
      IF(KL-9)40,50,46
   46 DO40I8=1,2
      JJ(8)=I8-1
      IF(KL-10)40,50,50
   50 CONTINUE
      NA=1
      DO80I=1,KL2
      II=KL2-I
      NA=NA+(2**II)*JJ(I)
   80 CONTINUE
C GEN. OF NB,NC,ND
      NB=NA+NODES/2
      NC=(2*NA)-1
      ND=2*NA
   81 FORMAT(5X,'NA NB NC ND ','III',5I5)
C FOR A PART. UPPER SEGMENT NODE NA THE DISTANCES LAC,LAD,LBC,LBD
C ARE EVALUATED.
      DO90I=1,NRAT
      E(I)=0.0
      DO100J=2,KL1
      K=KL-J
      E(I)=E(I)+TAP(I,J)*JJ(K)
  100 CONTINUE
      F(I)=E(I)+TAP(I,1)
      G(I)=E(I)+TAP(I,KL)
       H(I)=F(I)+TAP(I,KL)
      NEE=E(I)/2.0
      XEE=2*NEE
      NFF=F(I)/2.0
      XFF=2*NFF
      NGG=G(I)/2.0
      XGG=2*NGG
      NHH=H(I)/2.0
      XHH=2*NHH
      IF(E(I)-XEE)120,120,121
  120 NE(I)=0
      GO TO 131
  121 NE(I)=1
 131  IF(F(I)-XFF)122,122,123
  122 NF(I)=0
      GOTO132
  123 NF(I)=1
 132  IF(G(I)-XGG)124,124,125
  124 NG(I)=0
```

```
  150 CONTINUE
   90 CONTINUE
      LAC(NA)=0
      LAD(NA)=0
      LBC(NA)=0
      LBD(NA)=0
      DO110I=1,NRAT
        LAC(NA)=LAC(NA)+NE(I)
      LAD(NA)=LAD(NA)+NF(I)
      LBC(NA)=LBC(NA)+NG(I)
      LBD(NA)=LBD(NA)+NH(I)
  110 CONTINUE
        IF(NA.NE.NDBY2)GO TO 14
      TYPE21,(LAC(I),LAD(I),LBC(I),LBD(I),I=1,NDBY2)
   21 FORMAT(5X,'LAC',I5,'LAD',I5,'LBC',I5,'LBD',I5)
 14       CONTINUE
C WE KEEP A NOTE IN NZER(I) OF NODES NA, NB GIVING RISE TO
C ZERO DISTANCE SEGMENTS.
      IF(LAC(NA).EQ.0)GO TO 921
      IF(LAD(NA).NE.0)GO TO 922
  921 NZER(IZ)=NA
      IZ=IZ+1
  922 IF(LBC(NA).EQ.0)GO TO 923
      IF(LBD(NA).NE.0)GO TO 924
  923 NZER(IZ)=NB
      IZ=IZ+1
  924 CONTINUE
   40 CONTINUE
      IZ=IZ-1
C TO IN. MATRIX A
      DO30I=1,NODES
      DO30J=1,NTT
      A(I,J)=0.0
   30 CONTINUE
      A(1,3)=1.0
      TYPE946,(NZER(I),I=1,IZ)
  946 FORMAT(5X,'NZER',I5)
C BEGINNING THE RECURSIVE MAT. MULTPLN.
      DO1000IIT=1,NITER

      LNTT=(IIT+1)*NRAT
      DO999NA=1,NDBY2
      NB=NA+NDBY2
      ND=2*NA
      NC=ND-1
C  B=PI*A
      DO180I=1,LNTT
      IAC=I-LAC(NA)+NRAT
      IBC=I-LBC(NA)+NRAT
      IAD=I-LAD(NA)+NRAT
      IBD=I-LBD(NA)+NRAT
      B(NC,I)=A(NA,IAC)+A(NB,IBC)
      B(ND,I)=A(NA,IAD)+A(NB,IBD)
C NOTE THE CHANGE IN PATTERN OF MAT. MULTPLN. FOR DIFF. STAGES.
         IF(IIT.EQ.NTP1)B(NC,I)=0.0
          IF(IIT.LE.NTP1)GO TO 180
         IF(NA.NE.1)GO TO 180
         B(NC,I)=A(NB,IBC)
         B(ND,I)=A(NB,IBD)
  180 CONTINUE
```

```
      IF(ITI.LT.NTP1)GO TO 368
     DO360J=1,LNTT
      RVTI(J)=RVTT(J)+B(1,J)
     DO360I=2,NODES
     RASN(J)=RASN(J)+B(I,J)
 360 CONTINUE
      IF(ITI.EQ.NITER)GO TO 362
     DO361J=1,LNTT
      DO361I=2,NODES
      RPE(J)=RPE(J)+B(I,J)
 361 CONTINUE
362   CONTINUE
      IF(ITI.LE.NTP1)GO TO 366
     DO370J=1,LNTT
     JX=J+2
     DO370I=2,IZ
     IDUM=NZER(I)
     RPE(J)=RPE(J)-A(IDUM,JX)
 370 CONTINUE
      IF(ITI.NE.NITER)GO TO 368
      DO 390I=2,NODES
      DO 390J=1,LNTT
      RVTI(J)=RVTT(J)+B(I,J)
390   CONTINUE
368   CONTINUE
     LTT=LNTT+2
     DO380I=2,NODES
      DO380J=3,LTT
     JZ=J-2
     A(I,J)=B(1,JZ)
 380 CONTINUE
      IF(ITI.LE.NTP1)A(1,3)=1.0
C     TYPE 371,ITI
C     TYPE 369,((A(I,J),J=1,LTT),I=1,NODES)
      IF(ITI.NE.NITER)GO TO 1000

369   FORMAT(5X,'A(I,J)=',9F8.1)
     TYPE981,(RPE(I),I=1,LNTT)
      TYPE 983,(RVIT(I),I=1,LNTT)
 981 FORMAT(5X,'RPE',2E15.8)
     TYPE982,(RASN(I),I=1,LNTT)
 982 FORMAT(5X,'RASN',2E15.8)
371   FORMAT(5X,'THE NUMBER OF THE ITERATION SEEN='I4)
983   FORMAT(5X,'RVIT',2E15.8)
1000 CONTINUE
     STOP
     END
```

```
C THIS PROGRAM CAN BE USED  TO SIMULATE A
C SEQUENTIAL DETECTOR (ML CRITERION) FOR THE CASE OF
C CONVOLUTIONAL CODES OF ANY K.THIS PROGRAM AS IT IS
C IS FOR A RATE 1/2 C.CODE OF K=5.
C THE FOLL. IS THE LIST OF VARIABLES USED:-
C ERR(100) GIVES THE ERR. DISBN. AT THE VARIOUS
C NODE LEVELS.R IS THE REGISTER CONTAINING THE
C FUTURE RECD. SYMBOLS. RT IS THE REG. CONTAINING
C THE CURRENT FUTURE SYMBOL BEING EXAMINED.ISN
C GIVES THE NO. OF FUTURE SYMBOL BEING EXAMINED.
C ASN GIVES THE NODE LEVEL AT WHICH THE LAST DECN.
C WAS TAKEN.REG. D AND Z CONTAIN SURVIVOR METRICS
C DURING THE ACS PROCESS.ASN GIVES THE NO. OF
C TERMIN. AT THE VAR. NODE LEVELS. THRESH GIVES
C THE THRESH NEEDED AT THE VAR. NODE LEVELS.
C F AND H GIVE INF. REGARDING THE OUT. SYMBOLS
C CORRESS. TO SEGMENTS INTER CONNECTING TWO NODE
C LEVELS.THIS FOR A C.CODE OF K=5.XBUFX GIVES THE
C MAX. NO. OF SUCCESSIVE TERM. AT THE NT(TERMINATING)
C NODE LEVEL
C THE REG.M CONTAINS THE METRICS OF THE SURVIVORS
C STORED IN THE ACTIVE STORAGE SECTION. REG. MS
C THE METRICS STORED IN THE DORMANT STORAGE SECTION.
C MX IS A REG. CONTAINING THE METRICS OF THE SURV.
C DURING TRANSFERS ETC.
C XSN/SX= RATE OF THE CODE. THE STD. DEVN. OF THE
C NOISE IS GIVEN BY SIGMA.
C FUTURE RECD. SYMBOLS. RT IS THE REG. CONTAINING
C NERR GIVES THE NO. OF ERRORS IN THE DETN.
C TODATE. PE GIVES THE BIT ERROR PROB.
C COUNT GIVES THE NO. OF SYMBOLS DECODED.
C [illegible] DET. WHEN PRINTING OF OUT.
C IS DESIRED.
      DIMENSION ERR(100)
      DIMENSION X(20),R(30,10),RT(10),D(100,2),ASN(50),THRESH(50),
     A F(100),G(100),Z(100,2)
      DIMENSION DUML(2)
      REAL M(100)
      REAL MX(100),MS(100)
C INITIALIZING VARABLES
      DIMENSION X1(100),X2(100),MD(100),M1(100)
      DATA F(1),F(4),F(5),F(8),F(10),F(11),F(14),F(15),
     A G(1),G(2),G(3),G(4),G(13),G(14),G(15),G(16)/16*-1.0/
      DATA F(2),F(3),F(6),F(9),F(12),F(13),F(16),F(7),
     A G(5),G(6),G(7),G(8),G(9),G(10),G(11),G(12) /16*1.0/
      DATA [illegible],KSN,K/2,5,1,5/
      [illegible]=1.0
      [illegible]=1.0
      ISN=1
      [illegible]
      [illegible]
      [illegible]*2.5
      [illegible]=0.0
      [illegible]
      [illegible]=0.0**2
      [illegible]
      [illegible]=200.0
      [illegible]
      [illegible]=1.0
      DO 317 I=1,[illegible]
  317 ERR(I)=0.0
      DO [illegible]=1,[illegible]
     A ASN(I)=[illegible]
      [illegible]=([illegible])
```

```
      ZS=10.0/ZS
      SIGMA=SQRT(ZS)
 142    FORMAT(5X,'SNR='F6.2,5X,'SIGMA=',F6.2)
      H=SQRT(10.0)
      NTL1=NT-1
      NODES=2**(KSM*(K-1))
      NSEG=(2**KSM)*NODES
      NPASYM=KSM*(K-1)
      DO230I=1,NT
        THRESH(I)=330.0
 230  CONTINUE
      THRESH(NT)=1.0
       DO144I=1,NT
      ZT=THRESH(I)

        THRESH(I)=ALOG(ZT)
 144    IF(I.NE.NT)THRESH(I)=4.0

 145  FORMAT(5X,'VALUES OF LN TH',5E15.2)
       DO22I=1,NT
       DO22J=1,NX
         ZG1=RAN(DUM)
         ZG2=RAN(DUM)*2.0*3.14159
       ZG1=(-2.0)*ALOG(ZG1)
       ZG1=SQRT(ZG1)
       ZG2=COS(ZG2)
       GAUSS=ZG1*ZG2*SIGMA
       R(I,J)=-H+GAUSS
  22  CONTINUE
 220  FORMAT(5X,'R(5,2)',5E15.4)
         DO19J=1,2
         NS(J)=0
 19      CONTINUE
        DO1000TII=1,10000
        DO1000JJ=1,10000
        IF(XDEC.NE.0.0)GO TO 30
C JUST BEGAN DETN. PROCESS.
        IS[illegible]=1

C RECALLING DORMANT STORAGE.

        DO181J=1,NODES
         [illegible](I)=NS(J)
 181  CONTINUE
C LOOK AT 1ST FUTURE SYMBOL.
  51  DO36J=1,NTL1
        K=I+1
        DO36J=1,[illegible]
        R(I,J)=R(K,J)
  36  CONTINUE
C LOOK AT 1ST FUTURE SYMBOL
        DO10J=1,[illegible]
        RE(J)=R(1,J)
  10  CONTINUE
C GEN. A [illegible] RECD. SYMBOL.
        DO2[illegible]I=1,[illegible]
          [illegible]=RAN(DUM)
          ZG2=RAN(DUM)
          ZG2=2.0*3.14159*ZG2
        ZG1=(-2.0)*ALOG(ZG1)
        ZG1=SQRT(ZG1)
        ZG2=COS(ZG2)
        GAUSS=ZG1*ZG2*SIGMA
        R([illegible],J)=-H+GAUSS
```

```
      GOTO50
C CONTINUING THE DETN. PROCESS . THEREFORE
C LOOKING AT NEXT FUTURE SYMBOL.
   30 ISN=ISN+1
  234    FORMAT(I4)
      DO40J=1,N
      RT(J)=R(ISN,J)
   40 CONTINUE
C EVALUATING SEGMENT METRICS.
  50     CONTINUE
      DO70J=1,NODES
      D(J,2)=F(J)*RT(1)+G(J)*RT(2)
      D(J,1)=-D(J,2)
   70 CONTINUE
C ACS OF SURVIVOR METRICS AS WELL AS SURV.
C SYMBOLS. NOTE SURV. SYMBOLS HAVE TO
C BE FRESHLY GEN. DURING DETN. INV. 1ST
C FUTURE RECD. SYMBOL.
         DO177J=1,NODES
      DO177K=1,2
      Z( J,K)=M( J)+D( J,K)
  177 CONTINUE
      ND=NODES-1
      DO178J=1,ND,2
      L=(J+1)/2
      K=L+(NODES/2)
      MM=J+1
        IF(Z(L,1).GE.Z(K,1))GO TO 171
        M(J)=Z(L,1)
        IF(ISN.GT.1)X2(J)=X1(L)
        GO TO 172
  171   CONTINUE
        M(J)=Z(K,1)
        IF(ISN.GT.1)X2(J)=X1(K)
  172   CONTINUE
        IF(Z(L,2).GE.Z(K,2))GO TO 173
        M(MM)=Z(L,2)
        IF(ISN.GT.1)X2(MM)=X1(L)
        GO TO 178
  173   CONTINUE
        M(MM)=Z(K,2)
        IF(ISN.GT.1)X2(MM)=X1(K)
  178   CONTINUE
        [illegible]=1.0
C RESETTING THE METRICS IF THEY BECOME UNWIELDY.
        METRED=0
        DO842J=1,NODES
        IF(M(J).GT.1000.0)METRED=METRED+1
  842   CONTINUE
        IF(METRED.NE.NODES)GOTO 843
        DO844J=1,NODES
        M(J)=M(J)-1000.0
  844   CONTINUE
  843   CONTINUE
        DO 250 J=1,NODES
        X1(J)=X2(J)
  250   CONTINUE
        [illegible]=1
        [illegible]=1
        IF(ISN.NE.1)GO TO 890
C GEN. OF [illegible] MESSAGE SYMBOLS.
        [illegible]I=1,[illegible],2
        [illegible](I)=0.0
        [illegible](I)=[illegible](I)
```

```
          DO 892I=2,NODES,2
          X2(I)=1.0
          X3(I)=M(I)
892       CONTINUE
890       CONTINUE
          DO895I=1,NODES
          X1(I)=X2(I)
895       CONTINUE
          DO880I=1,NODES
          IF(X2(I).NE.0.0)GO TO 881
          M0(JD)=I
          JD=JD+1
          GO TO 880
881       CONTINUE
          M1(KD)=I
          KD=KD+1
880       CONTINUE
          M0MAX=JD-1
          M1MAX=KD-1
          TYPE*,(MX(I),I=1,NODES),NODES,MOMAX,M1MAX
          TYPE*,(M0(I),I=1,MOMAX),(M1(I),I=1,M1MAX),(DUML(I),I=1,2)
          CALL MEASE(DUML,NODES,MOMAX,M1MAX,M0,M1,MX)
        DUMLZ=DUML(1)
        DUML1=DUML(2)
          TYPE*,DUML1,DUMLZ
860       FORMAT(5X,'DUMLZ,DUML1=',F9.2,5X,F9.2)
C THE SEQ. DETECTION TEST BEGINS NOW.
               XLMA=2.0*(SIGMA**2)*THRESH(ISN)
        YLMA=XLMA/(2.0*H)
        Q=DUMLZ+YLMA
        IF(Q.GT.DUML1)GO TO 500
        X(KL)=0.0
        GO TO 800
  500 Q=DUML1+YLMA
        IF(Q.GT.DUMLZ)GO TO 1000
        X(KL)=1.0
C UPDATE RESULTS(STATISTICS)
C FINDING NO. OF SUCC. TERMN. AT NT TH NODE LEVEL.
  800 ASN(ISN)=ASN(ISN)+1.0
          IF(ISN.EQ.NT)GO TO 801
          IF(JSN.EQ.NT)GO TO 802
          CBUFX=0.0
          GO TO 810
802       IF(CBUFX.GT.XBUFX)XBUFX=CBUFX
          CBUFX=0.0
          GO TO 810
801       IF(JSN.EQ.NT)GO TO 803
          CBUFX=1.0
          GO TO 810
803       CBUFX      =CBUFX+1.0
810       CONTINUE

          JSN=ISN
        XDFC=0.0
          IF(COUNT.NE.PR)GO TO 951

          XUT=0.0
          TYPE206,(ASN(I),I=1,NT)
          AVE=0.0
          DO812I=1,NT
```

```
      PE=XBU/C[illegible]
      TYPE 813,AVE,COUNT,PE,XBUFX
      FORMAT(5X,'AVE=',F8.4,'COUNT=',F9.1,5X,'PE=',E9.2,
     1 5X,'XBUFX=',F6.1)
      FORMAT(5X,'SNR=',F5.1,5X,'SIGMA=',5X,F6.3,'LN A=',F6.1)
      TYPE 814,SNR,SIGMA,THRESH(1)
106 FORMAT(5X,'ASN(NT)',8F9.1)
      [illegible]
174 FORMAT(5X,'NERR',I5)
      TYPE 318,(ERR(I),I=1,NT)
318 FORMAT(5X,'ERR',10F5.1)
    PE=PR+NERR1
      COUNT=COUNT+1.0
    IF(COUNT.EQ.NENSTR)GO TO 1111
    IF(X(KL).EQ.0.0)GO TO 1000
    NERR=NERR+1
      ERR(ISN)=ERR(ISN)+1.0
000 CONTINUE
    TYPE 3,(ASN(I),I=1,NT)
  3 FORMAT(5X,'FINAL VALUES OF ASN(NT)',8F9.1)
    TYPE 175,NERR
175 FORMAT(5X,'NERR',I5)
11    CONTINUE
    STOP
    END
      SUBROUTINE LEAST(DUM,NPTS,MOMAX,M1MAX,M0,M1,MX)
      DIMENSION DUM(2),M0(100),M1(100)
      READ MX(100)
      TYPE *,(MX(I),I=1,NPTS)
      TYPE*,(MX(I),I=1,NPTS),NPTS,MOMAX,M1MAX
      TYPE *,(M0(I),I=1,MOMAX),(M1(I),I=1,M1MAX),(DUM(I),I=1,2)
5     FORMAT(5X,'DUM IN SUB=',E16.7)
      DUM(1)=100.0**2
      DUM(2)=100.0**2
      DO10I=1,MOMAX
      LL=M0(I)
      IF(MX(LL).LT.DUM(1))DUM(1)=MX(LL)
0     CONTINUE
      DO20I=1,M1MAX
      LL=M1(I)
      IF(MX(LL).LT.DUM(2))DUM(2)=MX(LL)
      CONTINUE
      RETURN
```

LIST OF REFERENCES

|AB-68| K.Abend, T.J.Harley, Jr., B.D.Fritchman and C, Gumacos, " On optimum receivers for channels having memory," IEEE. Trans. Inform. Theory (Corressp.) Vol. IT-14,pp.818-819,Nov.1968.

|AB-70| K.Abend and B.D.Fritchman,''Statistical detection for communication channels with intersymbol interference," Proc.IEEE ,Vol. 58,pp.779-785,May 19 70.

|AUS-67| M.E.Austin, "Decision feedback equalization for communication over dispersive channels, "M.I.T. Lincoln Lab.,Lexington,Mass.Tech.Rep. 437,Aug.1967; also Sc.D. dissertation;M.I.T.,Cambridge,Mass., May 1967.

|BOW-69| R.R.Bowen,"Bayesian decision proceedure for interfering digital signals," IEEE Trans.Inform. Theory(Corresp.), Vol. IT-15, pp. 506-507, July 1969.

|BTH-67| T.L.Booth,"Sequential machines and automata theory," New York, Wiley, 1967.

|CH-66| R.W.Chang and J.C.Hancock,"On receivers structures for channels having memory," IEEE Trans. Inform. Theory, Vol. IT-12,pp. 463-468, Oct.1966.

|DR-68| J.V.Di Franco and W.L.Rubin,"Radar Detection ," Prentice Hall Inc.,1968.

|FAN-63| R.M.Fano,"A heuristic discussion of probalistic decoding," IEEE Trans. Inform. Theory, Vol.IT-9,Apr. 1963,pp. 64-74.

|FOR-72| G.D.Forney,Jr.,"Maximum likelihood sequence estimation of digital signals in the presence of intersymbol interference," IEEE Trans. Inform. Theory, Vol. IT-18,pp. 363-378,1972 May.

|GOU-72| M.M.Goutmann, "Intersymbol Interference as a natural code," IEEE Trans. Commun. Technol. (Corresp.), Vol.COM-20, pp. 1033-1037, Oct. 1972.

|HJ-71| J.A.Heller and I.M.Jacobs," Viterbi decoding for satellite and space communication," IEEE Trans. Commun. Technol., Vol.COM-19, No.5,pp. 835-848, Oct. 1971.

|KOB-71| H.Kobayashi, "Application of probabilistic decoding to digital magnetic recording systems," IBM J.Res. Develop.,Vol.15, 1971.

|KOB-72| H.Kobayashi," Correlative level coding and maximum likelihood decoding,"IEEE Trans. Inform. Theory, Vol.IT-17,pp. 586-594, Sept. 1971.

|KW-71| T.J.Klein and J.K.Wolf,"On the use of channel introduced redundancy for error correction," IEEE. Trans.Commun. Technol., Vol.COM-19,pp.396-402, Aug. 1971.

|LAR-73| Larsen,"Short constraint length convolutional codes with maximum minimum free distance for rates 1/2,1/3,1/4," IEEE Trans. Inform. Theory, Vol.IT-19,May 1973.

|LUC-73| R.W.Lucky,"A survey of the communication theory literature : 1968-1973,"IEEE Trans. Inform. Theory, Vol.IT-19, pp. 725-740, Nov.1973.

|MAS-63| James L Massey, "Threshold decoding," M.I.T. Press , Cambridge, Mass.,1963.

|MB-55| J.J.Bussganj and D.Middleton,"Optimum sequential detection of signals in noise," IRE Trans. on Inform. Theory, Vol.IT-1, pp. 5-18,Dec.1955.

pp.363-378,May 1972.

|MON-71| P.Monsen,"Feedback equalization for fading dispersive channels," IEEE Trans. Inform. Theory, Vol.IT-17, pp. 55-65,Jan.1971.

|MON-74| P.Monsen,"Adaptive equalization of the slow fading channel, IEEE Trans. Commun. Technol., Vol.COM-22, No.8, pp.1064-1075, August 1974.

|NAR-78| N.C.Vyas, "Fading dispersive channel estimation and equalization using circulating filters, " M.Tech. Thesis, 78/219/MT, Indian Institute of Technology, Kanpur 1978.

|OMU-71| J.K.Omura, "Optimal receiver design for convolutional cods and channels with memory via control theoretical concepts," Inform. Sci.,Vol.3,pp.243-266, July 1971.

|STO-71| D.A.George, R.R.Bowen, and J.R.Storey, "An adaptive decision feedback equalizer, "IEEE Trans.Commun. Technol., Vol. COM-19,pp. 281-293,June 1971.

|TUF-63| D.W.Tufts, "Nyquist's problem - The joint optimization of transmitter and receiver in pulse amplitude modulation," Proc.IEEE, Vol. 53,pp. 248-260,Mar.1965.

|TUR-65| G.L.Turin, "Signal design for sequential detection systems with feedback," IEEE Trans. Inform. Theory, Vol.IT-11,pp. 401-408,July 1965.

|VB-75| G.Vinaya Babu, "A prototype decoder for short convolutional codes," M.Tech. Thesis, 75/216/MT, Indian Institute of Technology,Kanpur.

|VIT-67| A.J.Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," IEEE Trans. Inform. Theory, Vol.IT-13,pp.260-269, Apr. 1967.

|VIT-71| A.J.Viterbi,"Convolutional codes and their performance in communication systems,"IEEE Trans. on Commun. Technol.,Vol. COM-19,No.5,pp.751-772, Oct. 1971.

|WAL-47| A.Wald, "Sequential analysis," Wiley, New York, 1947.

|WAL-48| A.Wald and J.Wolfowitz, "Optimum character of the sequential probability ratio test," Ann. Math. Statistics, Vol.19, pp. 326- , 1948.

|WOZ-57| J.M.Wozencraft, "Sequential decoding for reliable communication," in 1957 IRE Nat. Conv. Record, Vol.5, pt. 2, pp.11-25.

****